Package 'WH'

June 20, 2025

Type Package

Title Enhanced Implementation of Whittaker-Henderson Smoothing

Version 2.0.0

Description An enhanced implementation of Whittaker-Henderson smoothing for the graduation of one-dimensional and two-dimensional actuarial tables used to quantify Life Insurance risks. 'WH' is based on the methods described in Biessy (2025) <doi:10.48550/arXiv.2306.06932>. Among other features, it generalizes the original smoothing algorithm to maximum likelihood estimation,
automatically selects the smoothing parameter(s) and extrapolates beyond the range of data.

License GPL (>= 3)

Encoding UTF-8

LazyData true

URL https://github.com/GuillaumeBiessy/WH

BugReports https://github.com/GuillaumeBiessy/WH/issues

Depends R (>= 4.2)

Imports Rcpp, stats

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

LinkingTo Rcpp

SystemRequirements LAPACK

Config/testthat/edition 3

RoxygenNote 7.3.2

VignetteBuilder knitr

Language en-US

NeedsCompilation yes

Author Guillaume Biessy [aut, cre, cph] (ORCID:
<https://orcid.org/0000-0003-3756-7345>)

Maintainer Guillaume Biessy <guillaume.biessy78@gmail.com>

Repository CRAN

Date/Publication 2025-06-19 23:20:02 UTC

Contents

`WH-package` *WH : Enhanced Implementation of Whittaker-Henderson Smoothing*

Description

An enhanced implementation of Whittaker-Henderson smoothing for the gradation of one-dimensional and two-dimensional actuarial tables used to quantify Life Insurance risks. `WH` is based on the methods described in Biessy (2025) doi:10.48550/arXiv.2306.06932. Among other features, it generalizes the original smoothing algorithm to maximum likelihood estimation, automatically selects the smoothing parameter(s) and extrapolates beyond the range of data.

Author(s)

Maintainer: Guillaume Biessy <guillaume.biessy78@gmail.com> (ORCID) [copyright holder]

See Also

Useful links:

- https://github.com/GuillaumeBiessy/WH
- Report bugs at https://github.com/GuillaumeBiessy/WH/issues

output_to_df *Store WH model fit results in a data.frame*

Description

Store WH model fit results in a data.frame

Usage

```
output_to_df(object, dim1 = "x", dim2 = "z")
```

Arguments

object	An object of class "WH_1d" or "WH_2d" returned by the WH() function
dim1	The (optional) name to be given to the first dimension
dim2	The (optional) name to be given to the second dimension

Value

A data.frame gathering information about the fitted and predicted values, the model variance, residuals and effective degrees of freedom...

plot.WH_1d *Plot 1D WH fit*

Description

Plot 1D WH fit

Usage

```
## S3 method for class 'WH_1d'
plot(x, what = "fit", trans, ...)
```

Arguments

x	An object of class "WH_1d" returned by the WH() function
what	What should be plotted. Should be one of fit (the default), res for residuals and edf for the effective degrees of freedom.
trans	An (optional) transformation to be applied to the data. By default the identity function
...	Not used

Value

A plot representing the desired element of the fit

Examples

```
d <- portfolio_mort$d
ec <- portfolio_mort$ec

WH(d, ec) |> plot()
WH(d, ec) |> plot("res")
WH(d, ec) |> plot("edf")
```

`plot.WH_2d` *Plot 2D WH fit*

Description

Plot 2D WH fit

Usage

```
## S3 method for class 'WH_2d'
plot(x, what = "y_hat", trans, ...)
```

Arguments

`x`	An object of class "WH_2d" returned by the `WH()` function
`what`	What should be plotted (y_hat, std_y_hat, res, edf)
`trans`	An (optional) transformation to be applied to the data
`...`	Not used

Value

A plot representing the given element of the fit...

Examples

```
d  <- portfolio_LTC$d
ec <- portfolio_LTC$ec

WH(d, ec) |> plot()
WH(d, ec) |> plot("std_y_hat")
```

portfolio_LTC *Aggregated Long-Term Care Dataset*

Description

Aggregated dataset built from a simulated long-term care portfolio

Usage

```
portfolio_LTC
```

Format

An synthetic aggregated dataset with death and exposure counts from a simulated long-term care portfolio with 100,000 contributors on a 20-year observation period (only deaths following long-term care claims are counted). The dataset is supplied as a list with two components :

d A matrix containing the portfolio number of observed deaths for each combination of age from 70 to 100 (excluded) and duration in LTC from 0 to 15 (excluded)

ec A matrix containing the portfolio central exposure in person-years for each combination age from 70 to 100 (excluded) and duration in LTC from 0 to 15 (excluded)

portfolio_mort *Aggregated Mortality Dataset*

Description

Aggregated dataset built from a simulated mortality portfolio

Usage

```
portfolio_mort
```

Format

An synthetic aggregated dataset with death and exposure counts from a simulated annuity portfolio with 100,000 contributors on a 20-year observation period. The dataset is supplied as a list with two components :

d A vector containing the portfolio number of observed deaths for each age from 50 to 95 (excluded)

ec A vector containing the portfolio central exposure in person-years for each age from 50 to 95 (excluded)

predict.WH_1d *Predict new values using a fitted 1D WH model*

Description

Extrapolate the model for new observations.

Usage

```
## S3 method for class 'WH_1d'
predict(object, newdata = NULL, ...)
```

Arguments

object	An object of class "WH_1d" returned by the WH() function
newdata	A vector containing the position of new observations. Observations from the fit will automatically be added to this, in the adequate order
...	Not used

Value

An object of class "WH_1d" with additional components for model prediction.

Examples

```
object <- WH(portfolio_mort$d, portfolio_mort$ec)
object_extra <- predict(object, newdata = 40:99)
plot(object_extra)
```

predict.WH_2d *Predict new values using a fitted 2D WH model*

Description

Extrapolate the model for new observations in a way that is consistent with the fitted values

Usage

```
## S3 method for class 'WH_2d'
predict(object, newdata = NULL, ...)
```

Arguments

object	An object of class "WH_2d" returned by the WH() function
newdata	A list containing two vectors indicating the new observation positions
...	Not used

Value

An object of class "WH_2d" with additional components for model prediction.

Examples

```
object <- WH(portfolio_LTC$d, portfolio_LTC$ec)
object_extra <- predict(object, newdata = list(age = 60:109, duration = 0:19))
plot(object_extra)
```

---

print.WH_1d *Display of 1D WH object*

Description

Display of 1D WH object

Usage

```
## S3 method for class 'WH_1d'
print(x, ...)
```

Arguments

x	An object of class "WH_1d" returned by the WH() function
...	Not used

Value

Invisibly returns x.

Examples

```
WH(portfolio_mort$d, portfolio_mort$ec)
```

print.WH_2d *Display of 2D WH object*

Description

Display of 2D WH object

Usage

```
## S3 method for class 'WH_2d'
print(x, ...)
```

Arguments

x	An object of class "WH_2d" returned by the WH() function
...	Not used

Value

Invisibly returns x.

Examples

```
WH(portfolio_LTC$d, portfolio_LTC$ec)
```

vcov.WH_1d *Compute variance-covariance matrix of fitted 1D WH model*

Description

The variance-covariance matrix may be useful in case confidence intervals are required for quantities derived from the fitted values.

Usage

```
## S3 method for class 'WH_1d'
vcov(object, pred = TRUE, ...)
```

Arguments

object	An object of class "WH_1d" returned by the WH() function
pred	Should the variance-covariance matrix include the extrapolated values as well (if any) ?
...	Not used

Value

The variance-covariance matrix for the fitted values

Examples

```
object <- WH(portfolio_mort$d, portfolio_mort$ec)
vcov(object)

object_extra <- predict(object, newdata = 40:99)
V <- vcov(object_extra)
```

`vcov.WH_2d` *Compute variance-covariance matrix of fitted 1D WH model*

Description

The variance-covariance matrix may be useful in case confidence intervals are required for quantities derived from the fitted values.

Usage

```
## S3 method for class 'WH_2d'
vcov(object, pred = TRUE, ...)
```

Arguments

`object`	An object of class "WH_2d" returned by the `WH()` function
`pred`	Should the variance-covariance matrix include the extrapolated values as well (if any) ?
`...`	Not used

Value

The variance-covariance matrix for the fitted values

Examples

```
object <- WH(portfolio_LTC$d, portfolio_LTC$ec)
V <- vcov(object)

object_extra <- predict(object, newdata = list(age = 60:109, duration = 0:19))
V <- vcov(object_extra)
```

WH *Whittaker-Henderson Smoothing*

Description

Main package function to apply Whittaker-Henderson Smoothing in a survival analysis framework. It takes as input two vectors / matrices of observed events and associated central exposure and estimate a smooth version of the log-hazard rate. Smoothing parameters may be supplied or automatically chosen according to a specific criterion such as `"REML"` (recommended), `"AIC"`, `"BIC"` or `"GCV"`. Whittaker-Henderson Smoothing may be applied in a full maximum likelihood framework (strongly recommended) or an asymptotic (approximate) Gaussian framework.

Usage

```
WH(d, ec, lambda = NULL, q = 2, criterion, reg, y, wt, verbose = 1, ...)
```

Arguments

`d`	Vector / matrix of observed events whose elements should be named.
`ec`	Vector / matrix of central exposure. The central exposure corresponds to the sum of the exposure period over the insured population. An individual experiencing an event of interest during the year will no longer be exposed afterwards and the exposure should be reduced accordingly.
`lambda`	Smoothing parameter. If missing, an optimization procedure will be used to find the optimal smoothing parameter.
`q`	Order of penalization. Polynoms of degrees `q - 1` are considered smooth and therefore unpenalized. The default of `2` should be suitable for most practical applications. Higher orders may cause numerical issues.
`criterion`	Criterion to be used for the selection of the optimal smoothing parameter. Default is `"REML"` which stands for restricted maximum likelihood. Other options include `"AIC"`, `"BIC"` and `"GCV"`.
`reg`	Should an approximate regression framework be used ? framework.
`y`	Optional vector of observations whose elements should be named. Used only in the regression framework and even in this case will be automatically computed from the `d` and `ec` arguments if those are supplied. May be useful when using Whittaker-Henderson smoothing outside of the survival analysis framework.
`wt`	Optional vector / matrix of weights. As for the observation vector / matrix `y`, used only in the regression framework and even in this case will be automatically computed if the `d` argument is supplied. May be useful when using Whittaker-Henderson smoothing outside of the survival analysis framework.
`verbose`	Integer between 0 and 3. Control the level of informations that will be printed on screen during fitting.
`...`	Additional parameters passed to the smoothing function called.

Value

An object of class `WH_1d` i.e. a list containing, among other things :

- `y` The observation vector/matrix, either supplied or computed as y = log(d) - log(ec)
- `y_hat` The vector/matrix of fitted value
- `std_y_hat` The vector/matrix of standard deviation associated with the fitted value
- `res` The vector/matrix of model deviance residuals
- `edf` The vector/matrix of effective degrees of freedom associated with each observation
- `diagnosis` A data.frame with one row containing the effective degrees of freedom of the model, the deviance of the fit as well as the AIC, BIC, GCV and REML criteria

Examples

```
d <- portfolio_mort$d
ec <- portfolio_mort$ec

y <- log(d / ec)
y[d == 0 | ec == 0] <- NA
wt <- d

# Maximum likelihood
WH(d, ec) # automatic smoothing parameter selection via REML
WH(d, ec, lambda = 1e2) # fixed smoothing parameter
WH(d, ec, criterion = "GCV") # alternative criterion for smoothing parameter selection

# Regression
WH(y = y, wt = wt) # regression framework is default when y is supplied
WH(d, ec, reg = TRUE, lambda = 1e2) # forces computation of y from d and ec
```

Index