

Package ‘WH’

August 30, 2024

Type Package

Title Enhanced Implementation of Whittaker-Henderson Smoothing

Version 1.1.2

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 (2023) <[doi:10.48550/arXiv.2306.06932](https://doi.org/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 stats

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

Config/testthat/edition 3

RoxygenNote 7.3.1

VignetteBuilder knitr

Language en-US

NeedsCompilation no

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

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

Repository CRAN

Date/Publication 2024-08-29 22:40:06 UTC

Contents

WH-package	2
output_to_df	3
plot.WH_1d	4
plot.WH_2d	4
portfolio_LTC	5
portfolio_mort	6
predict.WH_1d	6
predict.WH_2d	7
print.WH_1d	8
print.WH_2d	8
WH_1d	9
WH_2d	11

Index	15
--------------	-----------

WH-package	<i>WH : Enhanced Implementation of Whittaker-Henderson Smoothing</i>
------------	--

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 (2023) [doi:10.48550/arXiv.2306.06932](https://doi.org/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	<i>Provide WH model Fit Results as a Data.frame</i>
--------------	---

Description

Provide WH model Fit Results as a Data.frame

Usage

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

Arguments

object	An object of class "WH_1d" or "WH_2d" returned by one of the eponymous functions WH_1d() or WH_2d()
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 regrouping information about the fitted and predicted values, the model variance, residuals and effective degrees of freedom...

Examples

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

y <- log(d / ec)
y[d == 0] <- - 20
wt <- d

fit_1d <- WH_1d(d, ec)
output_to_df(fit_1d)

keep_age <- which(rowSums(portfolio_LTC$ec) > 5e2)
keep_duration <- which(colSums(portfolio_LTC$ec) > 1e3)

d <- portfolio_LTC$d[keep_age, keep_duration]
ec <- portfolio_LTC$ec[keep_age, keep_duration]

y <- log(d / ec) # observation vector
y[d == 0] <- - 20
wt <- d

# Maximum likelihood
fit_2d <- WH_2d(d, ec)
output_to_df(fit_2d)
```

plot.WH_1d

Plot Method for a Whittaker-Henderson Fit

Description

Plot Method for a Whittaker-Henderson Fit

Usage

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

Arguments

x	An object of class "WH_1d" returned by the <code>WH_1d()</code> function
what	What should be plotted. Should be one of <code>fit</code> (the model fit and standard deviation, the default), <code>res</code> for residuals and <code>edf</code> 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 from the fit

Examples

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

fit <- WH_1d(d, ec)
plot(fit)
plot(fit, "res")
plot(fit, "edf")
```

plot.WH_2d

Plot Method for a Whittaker-Henderson Fit

Description

Plot Method for a Whittaker-Henderson 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 <code>WH_2d()</code> function
what	Should be one of <code>y_hat</code> (the default) for model fit, <code>std_y_hat</code> for the associated standard deviation, <code>res</code> for residuals and <code>edf</code> 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 from the fit...

Examples

```
keep_age <- which(rowSums(portfolio_LTC$ec) > 5e2)
keep_duration <- which(colSums(portfolio_LTC$ec) > 1e3)

d <- portfolio_LTC$d[keep_age, keep_duration]
ec <- portfolio_LTC$ec[keep_age, keep_duration]

fit <- WH_2d(d, ec)
plot(fit)
plot(fit, "std_y_hat")
```

portfolio_LTC

Agregated Long-Term Care Dataset

Description

Agregated dataset built from a synthetic long-term care portfolio

Usage

```
portfolio_LTC
```

Format

A dataset obtaining by agregating the information from a fictive long-term care annuitant database with 5,000 annuitants over a 10-year observation period. The dataset is supplied as a list with two components :

exit A matrix containing the number of observed deaths for each combination of age and duration in LTC where at least one death has been observed

expo A matrix containing the associated central exposure in person-years for each combination of age and duration in LTC in `d`

portfolio_mort	<i>Agregated Mortality Dataset</i>
----------------	------------------------------------

Description

Agregated dataset built from a synthetic mortality portfolio

Usage

```
portfolio_mort
```

Format

A dataset containing the information from a simulated annuity portfolio with 100,000 contributors over a 10-year observation period. The dataset is supplied as a list with two components :

d A vector containing the number of observed deaths for ages where at least one death has been observed

ec A vector containing the associated central exposure in person-years for each age in **d**

predict.WH_1d	<i>Prediction for a Whittaker-Henderson Fit</i>
---------------	---

Description

Extrapolate the Whittaker-Henderson fit 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_1d() 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 `y_pred` and `std_y_pred` corresponding to the model predictions and associated standard deviations.

Examples

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

fit <- WH_1d(d, ec)
newdata = 18:99
pred <- predict(fit, newdata)
plot(pred)
```

predict.WH_2d

Prediction for a Whittaker-Henderson Fit

Description

Extrapolate the Whittaker-Henderson fit for new observations in a way that is consistent with the initial model fit.

Usage

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

Arguments

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

Value

An object of class "WH_2d" with additional components `y_pred` and `std_y_pred` corresponding to the model predictions and associated standard deviations.

Examples

```
keep_age <- which(rowSums(portfolio_LTC$ec) > 5e2)
keep_duration <- which(colSums(portfolio_LTC$ec) > 1e3)

d <- portfolio_LTC$d[keep_age, keep_duration]
ec <- portfolio_LTC$ec[keep_age, keep_duration]

fit <- WH_2d(d, ec)
newdata <- list(age = 50:99, duration = 0:19)
pred <- predict(fit, newdata)
plot(pred)
```

`print.WH_1d`*Print Method for a Whittaker-Henderson Fit*

Description

Print Method for a Whittaker-Henderson Fit

Usage

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

Arguments

<code>x</code>	An object of class "WH_1d" returned by the WH_1d() function
<code>...</code>	Not used

Value

Invisibly returns `x`.

Examples

```
d <- portfolio_mort$d  
ec <- portfolio_mort$ec  
  
y <- log(d / ec)  
y[d == 0] <- - 20  
wt <- d  
  
WH_1d(d, ec)
```

`print.WH_2d`*Print Method for a Whittaker-Henderson Fit*

Description

Print Method for a Whittaker-Henderson Fit

Usage

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


Arguments

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

Value

Invisibly returns x.

Examples

```
keep_age <- which(rowSums(portfolio_LTC$ec) > 5e2)
keep_duration <- which(colSums(portfolio_LTC$ec) > 1e3)

d <- portfolio_LTC$d[keep_age, keep_duration]
ec <- portfolio_LTC$ec[keep_age, keep_duration]

WH_2d(d, ec)
```

 WH_1d

1D Whittaker-Henderson Smoothing

Description

Main package function to apply Whittaker-Henderson smoothing in a one-dimensional survival analysis framework. It takes as input a vector of observed events and a vector of associated central exposure, both depending on a single covariate, and build a smooth version of the log-hazard rate. Smoothing parameters may be supplied or automatically chosen according to an adequate criterion such as "REML" (the default), "AIC", "BIC" or "GCV". Whittaker-Henderson may be applied in a full maximum likelihood framework (the default) or an approximate gaussian framework (the original).

Usage

```
WH_1d(
  d,
  ec,
  lambda,
  criterion,
  method,
  q = 2,
  framework,
  y,
  wt,
  quiet = FALSE,
  ...
)
```

Arguments

d	Vector of observed events, should have named elements.
ec	Vector of central exposure. The central exposure corresponds to the sum of the exposure duration over the insured population. An individual experiencing an event of interest during the year will no longer be exposed afterward and the exposure should be computed accordingly.
lambda	Smoothing parameter. If missing, an optimization procedure will be used to find the optimal smoothing parameter. If supplied, no optimal smoothing parameter search will take place unless the method argument is also supplied, in which case lambda will be used as the starting parameter for the optimization procedure.
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".
method	Method to be used to find the optimal smoothing parameter. Default to "fixed_lambda" if lambda is supplied, meaning no optimization is performed. Otherwise, default to the most reliable "outer" method based on the optimize function from package stats.
q	Order of penalization. Polynoms of degrees $q - 1$ are considered completely smooth and are therefore unpenalized. Should be left to the default of 2 for most practical applications.
framework	Default framework is "ml" which stands for maximum likelihood unless the y argument is also provided, in which case an "reg" or (approximate) regression framework is used.
y	Optional vector of observations whose elements should be named. Used only in the regression framework and if the d and ec arguments are missing (otherwise y is automatically computed from d and ec). May be useful when using Whittaker-Henderson smoothing outside of the survival analysis framework.
wt	Optional vector of weights. As for the observation vector y, used only in the regression framework and if the d and ec arguments are missing (otherwise wt is automatically set to d). May be useful when using Whittaker-Henderson smoothing outside of the survival analysis framework.
quiet	Should messages and warnings be silenced ? Default to FALSE, may be set to TRUE is the function is called repeatedly.
...	Additional parameters passed to the smoothing function called.

Value

An object of class WH_1d i.e. a list containing :

- d The inputed vector of observed events (if supplied as input)
- ec The inputed vector of central exposure (if supplied as input)
- y The observation vector, either supplied or computed as $y = \log(d) - \log(ec)$
- wt The inputed vector of weights, either supplied or computed as d
- y_hat The vector of values fitted using Whittaker-Henderson smoothing

- `std_y_hat` The vector of standard deviation associated with the fit
- `res` The vector of deviance residuals associated with the fit
- `edf_obs` The vector of effective degrees of freedom associated with each observation
- `edf_par` The vector of effective degrees of freedom associated with each eigenvector
- `diagnosis` A data.frame with one line containing the sum of effective degrees of freedom for the model, the deviance of the fit as well as the AIC, BIC, GCV and REML criteria
- `Psi` The variance-covariance matrix associated with the fit, which is required for the extrapolation step.
- `lambda` The smoothing parameter used.
- `p` The number of eigenvectors kept on each dimension if the rank reduction method is used (it should not in the one-dimensional case).
- `q` The supplied order for the penalization matrix.

Examples

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

y <- log(d / ec)
y[d == 0] <- - 20
wt <- d

# Maximum likelihood
WH_1d(d, ec, lambda = 1e2)
WH_1d(d, ec) # default outer iteration method based on the optimize function
WH_1d(d, ec, criterion = "GCV")
# alternative optimization criterion for smoothing parameter selection

# Regression
WH_1d(y = y, wt = wt, lambda = 1e2) # regression framework is default when y is supplied
WH_1d(d, ec, framework = "reg", lambda = 1e2)
# setting framework = "reg" forces computation of y from d and ec
```

Description

Main package function to apply Whittaker-Henderson smoothing in a bidimensional survival analysis framework. It takes as input a matrix of observed events and a matrix of associated central exposure, both depending on two covariates, and build a smooth version of the log-hazard rate. Smoothing parameters may be supplied or automatically chosen according to a specific criterion such as "REML" (the default), "AIC", "BIC" or "GCV". Whittaker-Henderson may be applied in a full maximum likelihood framework or an approximate gaussian framework. As Whittaker-Henderson smoothing relies on full-rank smoothers, computation time and memory usage in the bidimensional case may prove overwhelming and the function integrates a rank-reduction procedure to avoid such issues.

Usage

```

WH_2d(
  d,
  ec,
  lambda,
  criterion,
  method,
  max_dim = 200,
  p,
  q = c(2, 2),
  framework,
  y,
  wt,
  quiet = FALSE,
  ...
)

```

Arguments

d	Matrix of observed events, whose rows and columns must be named.
ec	Matrix of central exposure. The central exposure corresponds to the sum of the exposure duration over the insured population. An individual experiencing an event of interest during the year will no longer be exposed afterward and the exposure should be computed accordingly.
lambda	Smoothing parameter vector of size 2. If missing, an optimization procedure will be used to find the optimal smoothing parameter. If supplied, no optimal smoothing parameter search will take place unless the method argument is also supplied, in which case lambda will be used as the starting parameter for the optimization procedure.
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".
method	Method to be used to find the optimal smoothing parameter. Default to "fixed_lambda" if lambda is supplied, meaning no optimization is performed. Otherwise, default to "perf" which means the faster performance iteration method is used. The alternative "outer" method is safer but slower. Both those methods rely on the optim function from package stats.
max_dim	Number of parameters to be kept in the optimization problem. Default is 200. Values higher than 1000 may result in very high computation times and memory usage.
p	Optional vector of size 2. Maximum number of eigenvectors to keep on each dimension after performing the eigen decomposition of the penalization matrix. If missing, will be automatically computed so that the number of elements of the (square) matrices involved in the optimization problem remains lower than the max_dim argument

q	Order of penalization vector of size 2. Polynoms of degrees ($q[[1]] - 1, q[[2]] - 1$) are considered smooth and are therefore unpenalized. Should be left to the default of $c(2, 2)$ for most practical applications.
framework	Default framework is "ml" which stands for maximum likelihood unless the y argument is also provided, in which case an "reg" or (approximate) regression framework is used.
y	Optional matrix of observations whose rows and columns should be named. Used only in the regression framework and if the d and ec arguments are missing (otherwise y is automatically computed from d and ec). May be useful when using Whittaker-Henderson smoothing outside of the survival analysis framework.
wt	Optional matrix of weights. As for the observation vector y, Used only in the regression framework and if the d and ec arguments are missing (otherwise wt is set equal to d). May be useful when using Whittaker-Henderson smoothing outside of the survival analysis framework.
quiet	Should messages and warnings be silenced ? Default to FALSE, may be set to TRUE is the function is called repeatedly.
...	Additional parameters passed to the smoothing function called.

Value

An object of class WH_2d i.e. a list containing :

- d The inputed matrix of observed events (if supplied as input)
- ec The inputed matrix of central exposure (if supplied as input)
- y The observation matrix, either supplied or computed as $y = \log(d) - \log(ec)$
- wt The inputed matrix of weights, either supplied or set to d
- y_hat The matrix of values fitted using Whittaker-Henderson smoothing
- std_y_hat The matrix of standard deviations associated with the fit
- res The matrix of deviance residuals associated with the fit
- edf_obs The matrix of effective degrees of freedom associated with each observation
- edf_par The matrix of effective degrees of freedom associated with each eigenvector
- diagnosis A data.frame with one line containing the sum of effective degrees of freedom for the model, the deviance of the fit as well as the AIC, BIC, GCV and REML criteria
- Psi The variance-covariance matrix associated with the fit, which is required for the extrapolation step.
- lambda The vector of smoothing parameters used.
- p The number of eigenvectors kept on each dimension if the rank reduction method is used.
- q The supplied vector of orders for the penalization matrices.

Examples

```

keep_age <- which(rowSums(portfolio_LTC$ec) > 5e2)
keep_duration <- which(colSums(portfolio_LTC$ec) > 1e3)

d <- portfolio_LTC$d[keep_age, keep_duration]
ec <- portfolio_LTC$ec[keep_age, keep_duration]

y <- log(d / ec) # observation vector
y[d == 0] <- - 20
wt <- d

# Maximum likelihood
WH_2d(d, ec, lambda = c(1e2, 1e2))
WH_2d(d, ec) # performance iteration default method
WH_2d(d, ec, method = "outer") # slower but safer outer iteration method
WH_2d(d, ec, criterion = "GCV")
# alternative optimization criteria for smoothing parameter selection

# Regression
WH_2d(y = y, wt = wt, lambda = c(1e2, 1e2)) # regression framework is triggered when y is supplied
WH_2d(d, ec, framework = "reg", lambda = c(1e2, 1e2))
# setting framework = "reg" forces computation of y from d and ec

# Rank reduction
keep_age <- which(rowSums(portfolio_LTC$ec) > 1e2)
keep_duration <- which(colSums(portfolio_LTC$ec) > 1e2)

d <- portfolio_LTC$d[keep_age, keep_duration]
ec <- portfolio_LTC$ec[keep_age, keep_duration]

prod(dim(d)) # problem dimension is 627 !
WH_2d(d, ec)
# rank-reduction is used to find an approximate solution using 200 parameters

```

Index

* datasets

- portfolio_LTC, [5](#)
- portfolio_mort, [6](#)

output_to_df, [3](#)

- plot.WH_1d, [4](#)
- plot.WH_2d, [4](#)
- portfolio_LTC, [5](#)
- portfolio_mort, [6](#)
- predict.WH_1d, [6](#)
- predict.WH_2d, [7](#)
- print.WH_1d, [8](#)
- print.WH_2d, [8](#)

- WH (WH-package), [2](#)
- WH-package, [2](#)
- WH_1d, [9](#)
- WH_1d(), [3](#), [4](#), [6](#), [8](#)
- WH_2d, [11](#)
- WH_2d(), [3](#), [5](#), [7](#), [9](#)