

Package ‘StMoMo’

February 9, 2016

Title Stochastic Mortality Modelling

Version 0.3.1

Author Andres Villegas <andresmauriciovillegas@gmail.com>,
Pietro Millossovich <Pietro.Millossovich.1@city.ac.uk>,
Vladimir Kaishev <Vladimir.Kaishev.1@city.ac.uk>

Maintainer Andres Villegas <andresmauriciovillegas@gmail.com>

Description Implementation of the family of generalised age-period-cohort stochastic mortality models. This family of models encompasses many models proposed in the actuarial and demographic literature including the Lee-Carter model and the Cairns-Blake-Dowd model. It includes functions for fitting mortality models, analysing their goodness-of-fit and performing mortality projections and simulations.

URL <http://github.com/amvillegas/StMoMo>

BugReports <http://github.com/amvillegas/StMoMo/issues>

Imports MASS, rootSolve (>= 1.6.5.1), fanplot (>= 3.4), reshape2 (>= 1.4.1), fields (>= 8.2), RColorBrewer

Depends R (>= 3.2.0), gnm (>= 1.0), forecast(>= 6.1)

License GPL (>= 2)

LazyData true

Suggests knitr, xtable

VignetteBuilder knitr

RoxygenNote 5.0.1

NeedsCompilation no

Repository CRAN

Date/Publication 2016-02-09 20:53:52

R topics documented:

apc	2
bootstrap	4

bootstrap.fitStMoMo	4
cbd	6
coef.fitStMoMo	7
EWMaleData	8
fit	9
fit.StMoMo	9
fitted.fitStMoMo	12
forecast.fitStMoMo	13
forecast.mrwd	16
genWeightMat	17
lc	18
logLik.fitStMoMo	19
m6	20
m7	21
m8	22
mrwd	23
plot.bootStMoMo	24
plot.fitStMoMo	25
plot.forStMoMo	26
plot.resStMoMo	27
predict.fitStMoMo	28
print.bootStMoMo	30
print.fitStMoMo	31
print.forStMoMo	31
print.simStMoMo	32
print.StMoMo	32
residuals.fitStMoMo	33
rh	34
simulate.bootStMoMo	35
simulate.fitStMoMo	37
simulate.mrwd	41
StMoMo	41
Index	45

apc

Create an Age-Period-Cohort mortality model

Description

Utility function to initialise a StMoMo object representing an Age-Period-Cohort mortality model.

Usage

```
apc(link = c("log", "logit"))
```

Arguments

link defines the link function and random component associated with the mortality model. "log" would assume that deaths follow a Poisson distribution and use a log link while "logit" would assume that deaths follow a Binomial distribution and a logit link.

Details

The created model is either a log-Poisson or a logit-Binomial version of the classical age-period-cohort mortality model which has predictor structure

$$\eta_{xt} = \alpha_x + \kappa_t + \gamma_{t-x}.$$

To ensure identifiability we follow Cairns et al. (2009) and impose constraints

$$\sum_c \gamma_c = 0$$

and

$$\sum_c c\gamma_c = 0$$

Value

An object of class "StMoMo".

References

Cairns, A. J. G., Blake, D., Dowd, K., Coughlan, G. D., Epstein, D., Ong, A., & Balevich, I. (2009). A quantitative comparison of stochastic mortality models using data from England and Wales and the United States. *North American Actuarial Journal*, 13(1), 1-35.

See Also

[StMoMo](#), [rh](#)

Examples

```
APC <- apc()
wxt <- genWeightMat(EWMaleData$ages, EWMaleData$years, clip = 3)
APCfit <- fit(APC, Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
              ages = EWMaleData$ages, years = EWMaleData$years,
              wxt = wxt)
plot(APCfit, parametricbx = FALSE, nCol = 3)
```

bootstrap	<i>Generic method for bootstrapping a fitted Stochastic Mortality Model</i>
-----------	---

Description

bootstrap is a generic function for bootstrapping Stochastic Mortality Models. The function invokes particular methods which depend on the class of the first argument.

Usage

```
bootstrap(object, nBoot, ...)
```

Arguments

object	an object used to select a method. Typically of class <code>fitStMoMo</code> or an extension of this class.
nBoot	number of bootstrap samples to produce.
...	arguments to be passed to or from other methods.

Details

bootstrap is a generic function which means that new fitting strategies can be added for particular stochastic mortality models. See for instance [bootstrap.fitStMoMo](#).

bootstrap.fitStMoMo	<i>Bootstrap a fitted Stochastic Mortality Model</i>
---------------------	--

Description

Produce bootstrap parameters of a Stochastic Mortality Model to account for parameter uncertainty.

Usage

```
## S3 method for class 'fitStMoMo'
bootstrap(object, nBoot = 1, type = c("semiparametric",
  "residual"), deathType = c("observed", "fitted"), ...)
```

Arguments

object	an object of class <code>"fitStMoMo"</code> with the fitted parameters of a stochastic mortality model.
nBoot	number of bootstrap samples to produce.
type	type of bootstrapping approach to be applied. <code>"semiparametric"</code> (default) uses the assumed distribution of the deaths to generate bootstrap samples. <code>"residual"</code> resamples the deviance residuals of the model to generate bootstrap samples.

deathType	type of deaths to sample in the semiparametric bootstrap. "observed" (default) resamples the observed deaths. "fitted" resamples the fitted deaths. This parameter is only used if type is "semiparametric".
...	arguments to be passed to or from other methods.

Details

When type is "residual" the residual bootstrapping approach described in Renshaw and Haberman (2008) is applied, which is an adaptation of the approach of Koissi et al (2006). In the case of a "logit" link with Binomial responses the adaptation described in Debon et al, (2010, section 3) is used.

When type is "semiparametric" the semiparametric approach described in Brouhns et al.(2005) is used. In the case of a "logit" link with Binomial responses a suitable adaptation is applied. If deathType is "observed" then the observed deaths are used in the sampling as in Brouhns et al. (2005) while if deathType is "fitted" the fitted deaths are used in the sampling as in Renshaw and Haberman (2008).

Value

A list with class "bootStMoMo" with components:

bootParameters	a list of of length nBoot with the fitted parameters for each bootstrap replication.
model	the model fit that has been bootstrapped.
type	type of bootstrapping approach applied.
deathType	type of deaths sampled in case of semiparametric bootstrap.

References

- Brouhns, N., Denuit M., & Van Keilegom, I. (2005). Bootstrapping the Poisson log-bilinear model for mortality forecasting. *Scandinavian Actuarial Journal*, 2005(3), 212-224.
- Debon, A., Martinez-Ruiz, F., & Montes, F. (2010). A geostatistical approach for dynamic life tables: The effect of mortality on remaining lifetime and annuities. *Insurance: Mathematics and Economics*, 47(3), 327-336.
- Renshaw, A. E., & Haberman, S. (2008). On simulation-based approaches to risk measurement in mortality with specific reference to Poisson Lee-Carter modelling. *Insurance: Mathematics and Economics*, 42(2), 797-816.

See Also

[simulate.bootStMoMo](#), [plot.bootStMoMo](#)

Examples

```
#Long computing times
## Not run:
LCfit <- fit(lc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
            ages = EWMaleData$ages, years = EWMaleData$years)
```

```

LCResBoot <- bootstrap(LCfit, nBoot = 500, type = "residual")
plot(LCResBoot)
LCSEmiObsBoot <- bootstrap(LCfit, nBoot = 500, type = "semiparametric")
plot(LCSEmiObsBoot)
LCSEmiFitBoot <- bootstrap(LCfit, nBoot = 500, type = "semiparametric",
                           deathType = "fitted")
plot(LCSEmiFitBoot)

## End(Not run)

```

cbd

Create a Cairns-Blake-Dowd mortality model

Description

Utility function to initialise a StMoMo object representing a Cairns-Blake-Dowd mortality model.

Usage

```
cbd(link = c("logit", "log"))
```

Arguments

link defines the link function and random component associated with the mortality model. "log" would assume that deaths follow a Poisson distribution and use a log link while "logit" would assume that deaths follow a Binomial distribution and a logit link. Note that the default is the logit link.

Details

The created model is either a logit-Binomial or a log-Poisson version of the Cairns-Blake-Dowd mortality model which has predictor structure

$$\eta_{xt} = \kappa_t^{(1)} + (x - \bar{x})\kappa_t^{(2)},$$

where \bar{x} is the average age in the data.

Value

An object of class "StMoMo".

References

Cairns, A. J. G., Blake, D., & Dowd, K. (2006). A Two-Factor Model for Stochastic Mortality with Parameter Uncertainty: Theory and Calibration. *Journal of Risk and Insurance*, 73(4), 687-718.

See Also

[StMoMo](#), [m6](#), [m7](#), [m8](#)

Examples

```
CBD <- cbd()
Dxt <- EWMaleData$Dxt
Ext <- EWMaleData$Ext + 0.5 * EWMaleData$Dxt
CBDfit <- fit(CBD, Dxt = Dxt, Ext = Ext,
             ages = EWMaleData$ages, years = EWMaleData$years,
             ages.fit = 55:89)
plot(CBDfit, parametricbx = FALSE)
```

coef.fitStMoMo

Extract coefficients from a fitted Stochastic Mortality Model

Description

Extract coefficients from a fitted Stochastic Mortality Model

Usage

```
## S3 method for class 'fitStMoMo'
coef(object, ...)
```

Arguments

object	an object of class "fitStMoMo" with the fitted parameters of a stochastic mortality model.
...	other arguments.

Value

A list of model parameters with components:

ax	Vector with the fitted values of the static age function α_x . If the model does not have a static age function or failed to fit this is set to NULL.
bx	Matrix with the values of the period age-modulating functions $\beta_x^{(i)}, i = 1, \dots, N$. If the i -th age-modulating function is non-parametric (e.g., as in the Lee-Carter model) $bx[, i]$ contains the estimated values. If the model does not have any age-period terms (i.e. $N = 0$) or failed to fit this is set to NULL.
kt	Matrix with the values of the fitted period indexes $\kappa_t^{(i)}, i = 1, \dots, N$. $kt[i,]$ contains the estimated values of the i -th period index. If the model does not have any age-period terms (i.e. $N = 0$) or failed to fit this is set to NULL.
b0x	Vector with the values of the cohort age-modulating function $\beta_x^{(0)}$. If the age-modulating function is non-parametric $b0x$ contains the estimated values. If the model does not have a cohort effect or failed to fit this is set to NULL.
gc	Vector with the fitted cohort index γ_c . If the model does not have a cohort effect or failed to fit this is set to NULL.

Examples

```
APCfit <- fit(apc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
              ages = EWMaleData$ages, years = EWMaleData$years)
coef(APCfit)
```

EWMaleData

England and Wales male mortality data

Description

Age-specific deaths and exposures for England and Wales from the Human Mortality Database.

Usage

```
EWMaleData
```

Format

A list with the following components:

Dxt matrix of deaths data.

Ext matrix of exposures data (mid year population estimates).

ages vector of ages.

years vector of years.

Details

EWMaleData contains deaths and exposures for England and Wales males for the period 1961-2011 and for ages 0-100. Data taken from the Human Mortality Database on 5 November 2014.

Source

Human Mortality Database <http://www.mortality.org/>.

References

Human Mortality Database (2014). University of California, Berkeley (USA), and Max Planck Institute for Demographic Research (Germany). Available at www.mortality.org.

fit	<i>Generic for fitting a Stochastic Mortality Model</i>
-----	---

Description

`fit` is a generic function for fitting Stochastic Mortality Models. The function invokes particular methods which depend on the class of the first argument.

Usage

```
fit(object, ...)
```

Arguments

<code>object</code>	an object used to select a method. Typically of class <code>StMoMo</code> or an extension of this class.
<code>...</code>	arguments to be passed to or from other methods.

Details

`fit` is a generic function which means that new fitting strategies can be added for particular stochastic mortality models. See for instance [fit.StMoMo](#).

<code>fit.StMoMo</code>	<i>Fit a Stochastic Mortality Model</i>
-------------------------	---

Description

Fit a Stochastic Mortality Model to a given data set. The fitting is done using package `gnm`.

Usage

```
## S3 method for class 'StMoMo'
fit(object, Dxt, Ext, ages = 1:nrow(Dxt),
     years = 1:ncol(Dxt), ages.fit = ages, years.fit = years, oxt = NULL,
     wxt = NULL, start.ax = NULL, start.bx = NULL, start.kt = NULL,
     start.b0x = NULL, start.gc = NULL, verbose = TRUE, ...)
```

Arguments

<code>object</code>	an object of class "StMoMo" defining the stochastic mortality model.
<code>Dxt</code>	matrix of deaths data.
<code>Ext</code>	matrix of observed exposures of the same dimension of <code>Dxt</code> .
<code>ages</code>	vector of ages corresponding to rows of <code>Dxt</code> and <code>Ext</code> .
<code>years</code>	vector of years corresponding to rows of <code>Dxt</code> and <code>Ext</code> .
<code>ages.fit</code>	optional vector of ages to include in the fit. Must be a subset of <code>ages</code> .
<code>years.fit</code>	optional vector of years to include in the fit. Must be a subset of <code>years</code> .
<code>oxt</code>	optional matrix/vector or scalar of known offset to be used in fitting the model. This can be used to specify any a priori known component to be added to the predictor during fitting.
<code>wxt</code>	optional matrix of 0-1 weights to be used in the fitting process. This can be used, for instance, to zero weight some cohorts in the data. See genWeightMat which is a helper function for defining weighting matrices.
<code>start.ax</code>	optional vector with starting values for α_x .
<code>start.bx</code>	optional matrix with starting values for $\beta_x^{(i)}$.
<code>start.kt</code>	optional matrix with starting values for $\kappa_t^{(i)}$.
<code>start.b0x</code>	optional vector with starting values for $\beta_x^{(0)}$.
<code>start.gc</code>	optional vector with starting values for γ_c .
<code>verbose</code>	a logical value. If TRUE progress indicators are printed as the model is fitted. Set <code>verbose = FALSE</code> to silent the fitting and avoid progress messages.
<code>...</code>	arguments to be passed to or from other methods. This can be used to control the fitting parameters of gnm . See gnm .

Details

Fitting is done using function [gnm](#) within package [gnm](#). This is equivalent to minimising (maximising) the deviance (log-likelihood) of the model. Ages and years in the data should be of type numeric. Data points with zero exposure are assigned a zero weight and are ignored in the fitting process. Similarly, NA are assigned a zero weight and ignored in the fitting process. Parameter estimates can be plotted using function [plot.fitStMoMo](#).

Value

A list with class "fitStMoMo" with components:

<code>model</code>	the object of class "StMoMo" defining the fitted stochastic mortality model.
<code>ax</code>	vector with the fitted values of the static age function α_x . If the model does not have a static age function or failed to fit this is set to NULL.
<code>bx</code>	matrix with the values of the period age-modulating functions $\beta_x^{(i)}, i = 1, \dots, N$. If the i -th age-modulating function is non-parametric (e.g. as in the Lee-Carter model) <code>bx[, i]</code> contains the estimated values. If the model does not have any age-period terms (i.e. $N = 0$) or failed to fit this is set to NULL.

kt	matrix with the values of the fitted period indexes $\kappa_t^{(i)}, i = 1, \dots, N$. kt[i,] contains the estimated values of the i -th period index. If the model does not have any age-period terms (i.e. $N = 0$) or failed to fit this is set to NULL.
b0x	vector with the values of the cohort age-modulating function $\beta_x^{(0)}$. If the age-modulating function is non-parametric b0x contains the estimated values. If the model does not have a cohort effect or failed to fit this is set to NULL.
gc	vector with the fitted cohort index γ_c . If the model does not have a cohort effect or failed to fit this is set to NULL.
Dxt	matrix of deaths used in the fitting.
Ext	matrix of exposures used in the fitting.
oxt	matrix of known offset values used in the fitting.
wxt	matrix of 0-1 weights used in the fitting.
ages	vector of ages used in the fitting.
years	vector of years used in the fitting.
cohorts	vector of cohorts used in the fitting.
fittingModel	output from the call to gnm used to fit the model. If the fitting failed to converge this is set to NULL.
loglik	log-likelihood of the model. If the fitting failed to converge this is set to NULL.
deviance	deviance of the model. If the fitting failed to converge this is set to NULL.
npar	effective number of parameters in the model. If the fitting failed to converge this is set to NULL.
nobs	number of observations in the model fit. If the fitting failed to converge this is set to NULL.
fail	TRUE if a model could not be fitted and FALSE otherwise.
conv	TRUE if the model fitting converged and FALSE if it didn't.

@seealso [genWeightMat](#), [plot.fitStMoMo](#)

Examples

```
# CBD model only to older ages
CBDfit <- fit(cbd(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
             ages = EWMaleData$ages, years = EWMaleData$years,
             ages.fit = 55:89)
plot(CBDfit, parametricbx = FALSE)

# APC model weighting out the 3 first and last cohorts
wxt <- genWeightMat(EWMaleData$ages, EWMaleData$years, clip = 3)
APCfit <- fit(apc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
             ages = EWMaleData$ages, years = EWMaleData$years,
             wxt = wxt)
plot(APCfit, parametricbx = FALSE, nCol = 3)
```

```

# Set verbose = FALSE for silent fitting
APCfit <- fit(apc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
             ages = EWMaleData$ages, years = EWMaleData$years,
             wxt = wxt, verbose = FALSE)

## Not run:
# Poisson Lee-Carter model with the static age function set to
# the mean over time of the log-death rates
constLCfix_ax <- function(ax, bx, kt, b0x, gc, wxt, ages){
  c1 <- sum(bx, na.rm = TRUE)
  bx <- bx / c1
  kt <- kt * c1
  list(ax = ax, bx = bx, kt = kt, b0x = b0x, gc = gc)
}
LCfix_ax <- StMoMo(link = "log", staticAgeFun = FALSE,
                  periodAgeFun = "NP", constFun = constLCfix_ax)
LCfix_axfit <- fit(LCfix_ax, Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
                  ages = EWMaleData$ages, years = EWMaleData$years,
                  oxt = rowMeans(log(EWMaleData$Dxt / EWMaleData$Ext)))
plot(LCfix_axfit)

## End(Not run)

```

fitted.fitStMoMo

Compute fitted values for a Stochastic Mortality Model

Description

Returns fitted values for the data used in fitting a Stochastic Mortality Model.

Usage

```

## S3 method for class 'fitStMoMo'
fitted(object, type = c("link", "rates", "deaths"), ...)

```

Arguments

object	an object of class "fitStMoMo" with the fitted parameters of a stochastic mortality model.
type	the type of the fitted values that should be returned. The alternatives are "link"(default), "rates", and "deaths".
...	other arguments.

Value

A matrix with the fitted values.

Examples

```
LCfit <- fit(lc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
            ages = EWMaleData$ages, years = EWMaleData$years,
            ages.fit = 55:89)
matplot(LCfit$ages, fitted(LCfit), type = "l", lty = 1,
        col = rainbow(length(LCfit$years)), xlab = "year",
        ylab = "log death rate", main = "Fitted rates")

qxthat <- fitted(LCfit, type = "rates")
qxt <- LCfit$Dxt / LCfit$Ext
plot(LCfit$years, qxt["65", ], xlab = "year", ylab = "death rate",
     main = "fitted vs. observed rates at age 65")
lines(LCfit$years, qxthat["65", ])
```

forecast.fitStMoMo	<i>Forecast mortality rates using a Stochastic Mortality Model</i>
--------------------	--

Description

Forecast mortality rates using a Stochastic Mortality Model fit. The period indexes $\kappa_t^{(i)}, i = 1, \dots, N$, are forecasted using a Multivariate Random Walk with Drift (MRWD). The cohort index γ_{t-x} is forecasted using an ARIMA(p, d, q). By default an ARIMA(1, 1, 0) with a constant is used.

Usage

```
## S3 method for class 'fitStMoMo'
forecast(object, h = 50, level = 95, oxt = NULL,
         gc.order = c(1, 1, 0), gc.include.constant = TRUE, jumpchoice = c("fit",
         "actual"), kt.lookback = NULL, gc.lookback = NULL, ...)
```

Arguments

object	an object of class "fitStMoMo" with the fitted parameters of a stochastic mortality model.
h	number of years ahead to forecast.
level	confidence level for prediction intervals of the period and cohort indices.
oxt	optional matrix/vector or scalar of known offset to be added in the forecasting. This can be used to specify any a priori known component to be added to the forecasted predictor.
gc.order	a specification of the ARIMA model: the three components (p, d, q) are the AR order, the degree of differencing, and the MA order. The default is an ARIMA(1, 1, 0).
gc.include.constant	a logical value indicating if the ARIMA model should include a constant value. The default is TRUE.

jumpchoice	option to select the jump-off rates, i.e. the rates from the final year of observation, to use in projections of mortality rates. "fit"(default) uses the fitted rates and "actual" uses the actual rates from the final year.
kt.lookback	optional argument to specify the look-back window to use in the estimation of the MRWD for period indexes. By default all the estimated values are used in estimating the MRWD. If kt.lookback is provided then the last kt.lookback years of $\kappa_t^{(i)}, i = 1, \dots, N$, are used.
gc.lookback	optional argument to specify the look-back window to use in the estimation of the ARIMA model for the cohort effect. By default all the estimated values are used in estimating the ARIMA model. If gc.lookback is provided then the last gc.lookback years of γ_{t-x} are used.
...	other arguments.

Details

Fitting and forecasting of the Multivariate Random Walk with Drift for the period indexes use the function `mrwd`. Fitting and forecasting of the ARIMA model for the cohort index is done with function `Arima` from package **forecast**. See the latter function for further details on input arguments `gc.order` and `gc.include.constant`.

Note that in some cases forecast of the cohort effects may be needed for a horizon longer than `h`. This is the case when in the fitted model the most recent cohorts have been zero weighted. The forecasted cohorts can be seen in `gc.f$cohorts`.

Value

A list of class "forStMoMo" with components:

rates	a matrix with the point forecast of the rates.
ages	vector of ages corresponding to the rows of rates.
years	vector of years for which a forecast has been produced. This corresponds to the columns of rates.
kt.f	forecasts of period indexes of the model. This is a list with the model fitted to κ_t ; the mean(central) forecast, the lower and upper limits of the prediction interval; the confidence level associated with the prediction interval; and the years for which a forecast was produced. If the model does not have any age-period terms (i.e. $N = 0$) this is set to NULL.
gc.f	forecasts of cohort index of the model. This is a list with the model fitted to γ_c ; the mean(point) forecast, the lower and upper limits of the prediction interval; the confidence level associated with the prediction interval; and the cohorts for which a forecast was produced. If the mortality model does not have a cohort effect this is set to NULL.
oxt.f	the offset used in the forecast.
fitted	a matrix with the fitted in-sample rates of the model for the years for which the mortality model was fitted.
model	the model fit from which the forecast was produced.
jumpchoice	Jump-off method used in the forecast.

Examples

```

#Lee-Carter
LCfit <- fit(lc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
             ages = EWMaleData$ages, years = EWMaleData$years,
             ages.fit = 55:89)
LCfor <- forecast(LCfit)
plot(LCfor)

#CBD
CBDfit <- fit(cbd(),Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
             ages = EWMaleData$ages, years = EWMaleData$years,
             ages.fit = 55:89)
CBDfor <- forecast(CBDfit)
plot(CBDfor, parametricbx = FALSE)

#APC: Compare forecast with different models for the cohort index
wxt <- genWeightMat(55:89, EWMaleData$years, clip = 3)
APCfit <- fit(apc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
             ages = EWMaleData$ages, years = EWMaleData$years,
             ages.fit = 55:89, wxt = wxt)
APCfor1 <- forecast(APCfit)
plot(APCfor1, parametricbx = FALSE, nCol = 3)
APCfor2 <- forecast(APCfit, gc.order = c(0, 2, 2))
plot(APCfor2, only.gc = TRUE)
plot(c(APCfit$years, APCfor1$years),
     cbind(APCfor1$fitted, APCfor1$rates)["65", ],
     type = "l", xlab = "year", ylab = "Mortality rate at age 65",
     main = "Forecasts with different models for gc")
lines(APCfor2$years, APCfor2$rates["65", ], col = "blue")
points(APCfit$years, (APCfit$Dxt / APCfit$Ext)["65", ], pch = 19)

#Compare Lee-Carter forecast using:
# 1. Fitted jump-off rates and all history for kt
# 2. Actual jump-off rates and all history for kt
# 3. Fitted jump-off rates and only history for
#    the past 30 years of kt (i.e 1982-2011)

LCfor1 <- forecast(LCfit)
LCfor2 <- forecast(LCfit, jumpchoice = "actual")
LCfor3 <- forecast(LCfit, kt.lookback = 30)

plot(LCfit$years, (LCfit$Dxt / LCfit$Ext)["60", ],
     xlim = range(LCfit$years, LCfor1$years),
     ylim = range((LCfit$Dxt / LCfit$Ext)["60", ], LCfor1$rates["60", ],
                  LCfor2$rates["60", ], LCfor3$rates["60", ]),
     type = "p", xlab = "year", ylab = "rate",
     main = "Lee-Carter: Forecast of mortality rates at age 60")
lines(LCfit$years, fitted(LCfit, type = "rates")["60", ])
lines(LCfor1$years, LCfor1$rates["60", ], lty = 2)
lines(LCfor2$years, LCfor2$rates["60", ], lty = 3, col = "blue")
lines(LCfor3$years, LCfor3$rates["60", ], lty = 4, col = "red")
legend("topright", legend = c("Fitted jump-off", "Actual jump-off",

```

```
"Fitted jump-off, 30 year look-back"),
lty = 1:3, col = c("black", "blue", "red"))
```

forecast.mrwd

Forecast a Multivariate Random Walk with Drift

Description

Returns forecasts and other information for a Multivariate Random Walk with Drift model.

Usage

```
## S3 method for class 'mrwd'
forecast(object, h = 10, level = c(80, 95), fan = FALSE,
...)
```

Arguments

object	an object of class "mrwd".
h	Number of periods for forecasting.
level	confidence level for prediction intervals.
fan	if TRUE, level is set to seq(50, 99, by = 1). This is suitable for fan plots.
...	other arguments.

Value

An object of class "mrwdForecast" with components:

model	a list containing information about the fitted model.
mean	array with the central forecast.
lower	three dimensional array with lower limits for prediction intervals.
upper	three dimensional array with upper limits for prediction intervals.
level	the confidence values associated with the prediction intervals.
@export	

genWeightMat	<i>Generate weight matrix</i>
--------------	-------------------------------

Description

Generates a weight matrix given a group of ages and years and a set of cohorts which are to be given zero weight. This is useful for excluding some data points when fitting a Stochastic Mortality Model (see [fit.StMoMo](#)).

Usage

```
genWeightMat(ages, years, clip = 0, zeroCohorts = NULL)
```

Arguments

ages	vector of ages.
years	vector of years.
clip	number of cohorts in the boundary to assign a zero weight. This can be used to zero weigh some of the first and last cohorts in the data.
zeroCohorts	other cohort for which a zero weight is to be assigned.

Value

A 0-1 matrix with 0 for the zero-weighted cohorts.

See Also

[fit.StMoMo](#)

Examples

```
#Zero-weight the first three and last three cohorts
wxt1 <- genWeightMat(55:89, EWMaleData$years, clip = 3)
APCfit1 <- fit(apc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
              ages = EWMaleData$ages, years = EWMaleData$years,
              ages.fit = 55:89, wxt = wxt1)
plot(APCfit1, parametricbx = FALSE, nCol = 3)

#Also Zero-weight the 1886 cohort
wxt2 <- genWeightMat(55:89, EWMaleData$years, clip = 3,
                    zeroCohorts = 1886)
APCfit2 <- fit(apc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
              ages = EWMaleData$ages, years = EWMaleData$years,
              ages.fit = 55:89, wxt = wxt2)
plot(APCfit2, parametricbx = FALSE, nCol = 3)
```

lc	Create a Lee-Carter model
----	---------------------------

Description

Utility function to initialise a StMoMo object representing a Lee-Carter model.

Usage

```
lc(link = c("log", "logit"), const = c("sum", "last", "first"))
```

Arguments

link	defines the link function and random component associated with the mortality model. "log" would assume that deaths follow a Poisson distribution and use a log link while "logit" would assume that deaths follow a Binomial distribution and a logit link.
const	defines the constraint to impose to the period index of the model to ensure identifiability. The alternatives are "sum"(default), "last" and "first" which apply constraints $\sum_t \kappa_t = 0$, $\kappa_n = 0$ and $\kappa_1 = 0$, respectively.

Details

The created model is either a log-Poisson (see Brouhns et al (2002)) or a logit-Binomial version of the Lee-Carter model which has predictor structure

$$\eta_{xt} = \alpha_x + \beta_x \kappa_t.$$

To ensure identifiability one of the following constraints is imposed

$$\sum_t \kappa_t = 0, \kappa_1 = 0, \kappa_n = 0$$

depending on the value of const, and

$$\sum_x \beta_x = 1.$$

Value

An object of class "StMoMo".

References

Brouhns, N., Denuit, M., & Vermunt, J. K. (2002). A Poisson log-bilinear regression approach to the construction of projected lifetables. *Insurance: Mathematics and Economics*, 31(3), 373-393.

Lee, R. D., & Carter, L. R. (1992). Modeling and forecasting U.S. mortality. *Journal of the American Statistical Association*, 87(419), 659-671.

See Also[StMoMo](#)**Examples**

```
#sum(kt) = 0 and log link
LC1 <- lc()
LCfit1<-fit(LC1, Dxt = EWMaleData$Dxt,Ext = EWMaleData$Ext,
            ages = EWMaleData$ages, years = EWMaleData$years,
            ages.fit = 55:89)
plot(LCfit1)

#kt[1] = 0 and log link
LC2 <- lc(const = "first")
LCfit2<-fit(LC2, Dxt = EWMaleData$Dxt,Ext = EWMaleData$Ext,
            ages = EWMaleData$ages, years = EWMaleData$years,
            ages.fit = 55:89)
plot(LCfit2)

#kt[n] = 0 and logit link
LC3 <- lc("logit", "last")
LCfit3<-fit(LC3, Dxt = EWMaleData$Dxt,Ext = EWMaleData$Ext,
            ages = EWMaleData$ages, years = EWMaleData$years,
            ages.fit = 55:89)
plot(LCfit3)
```

logLik.fitStMoMo

*Log-Likelihood of a fitStMoMo object***Description**

Returns the log-likelihood of a fitted Stochastic Mortality Model.

Usage

```
## S3 method for class 'fitStMoMo'
logLik(object, ...)
```

Arguments

object	an object of class fitStMoMo representing a Stochastic Mortality Model fitted to some data.
...	other arguments.

Value

The log-likelihood of the fitted model.

m6	<i>Create an M6 type extension of the Cairns-Blake-Dowd mortality model</i>
----	---

Description

Utility function to initialise a StMoMo object representing the M6 (CBD with cohorts) extension of the Cairns-Blake-Dowd mortality model introduced in Cairns et al (2009).

Usage

```
m6(link = c("logit", "log"))
```

Arguments

link	defines the link function and random component associated with the mortality model. "log" would assume that deaths follow a Poisson distribution and use a log link while "logit" would assume that deaths follow a Binomial distribution and a logit link. Note that the default is the logit link.
------	--

Details

The created model is either a logit-Binomial or a log-Poisson version of the M6 model which has predictor structure

$$\eta_{xt} = \kappa_t^{(1)} + (x - \bar{x})\kappa_t^{(2)} + \gamma_{t-x},$$

where \bar{x} is the average age in the data.

Identifiability of the model is accomplished by applying parameters constraints

$$\sum_c \gamma_c = 0, \sum_c c\gamma_c = 0$$

which ensure that the cohort effect fluctuates around zero and has no linear trend. These constraints are applied using the strategy discussed in Appendix A of Haberman and Renshaw (2011).

Value

An object of class "StMoMo".

References

Cairns, A. J. G., Blake, D., Dowd, K., Coughlan, G. D., Epstein, D., Ong, A., & Balevich, I. (2009). A quantitative comparison of stochastic mortality models using data from England and Wales and the United States. *North American Actuarial Journal*, 13(1), 1-35.

Haberman, S., & Renshaw, A. (2011). A comparative study of parametric mortality projection models. *Insurance: Mathematics and Economics*, 48(1), 35-55.

See Also

[StMoMo](#), [cbd](#), [m7](#), [m8](#)

Examples

```
M6 <- m6()
Dxt <- EWMaleData$Dxt
Ext <- EWMaleData$Ext + 0.5 * EWMaleData$Dxt
wxt <- genWeightMat(55:89, EWMaleData$years, clip = 3)
M6fit <- fit(M6, Dxt = Dxt, Ext = Ext, ages = EWMaleData$ages,
             years = EWMaleData$years, ages.fit = 55:89)
plot(M6fit, parametricbx = FALSE)
```

m7	<i>Create an M7 type extension of the Cairns-Blake-Dowd mortality model</i>
----	---

Description

Utility function to initialise a StMoMo object representing the M7 extension of the Cairns-Blake-Dowd mortality model introduced in Cairns et al (2009).

Usage

```
m7(link = c("logit", "log"))
```

Arguments

link defines the link function and random component associated with the mortality model. "log" would assume that deaths follow a Poisson distribution and use a log link while "logit" would assume that deaths follow a Binomial distribution and a logit link. Note that the default is the logit link.

Details

The created model is either a logit-Binomial or a log-Poisson version of the M7 model which has predictor structure

$$\eta_{xt} = \kappa_t^{(1)} + (x - \bar{x})\kappa_t^{(2)} + ((x - \bar{x})^2 - \hat{\sigma}_x^2)\kappa_t^{(2)} + \gamma_{t-x},$$

where \bar{x} is the average age in the data and $\hat{\sigma}_x^2$ is the average value of $(x - \bar{x})^2$.

Identifiability of the model is accomplished by applying parameters constraints

$$\sum_c \gamma_c = 0, \sum_c c\gamma_c = 0, \sum_c c^2\gamma_c = 0$$

which ensure that the cohort effect fluctuates around zero and has no linear or quadratic trend. These constraints are applied using the strategy discussed in Appendix A of Haberman and Renshaw (2011).

Value

An object of class "StMoMo".

References

Cairns, A. J. G., Blake, D., Dowd, K., Coughlan, G. D., Epstein, D., Ong, A., & Balevich, I. (2009). A quantitative comparison of stochastic mortality models using data from England and Wales and the United States. *North American Actuarial Journal*, 13(1), 1-35.

Haberman, S., & Renshaw, A. (2011). A comparative study of parametric mortality projection models. *Insurance: Mathematics and Economics*, 48(1), 35-55.

See Also

[StMoMo](#), [cbd](#), [m6](#), [m8](#)

Examples

```
M7 <- m7()
Dxt <- EWMaleData$Dxt
Ext <- EWMaleData$Ext + 0.5 * EWMaleData$Dxt
wxt <- genWeightMat(55:89, EWMaleData$years, clip = 3)
M7fit <- fit(M7, Dxt = Dxt, Ext = Ext, ages = EWMaleData$ages,
             years = EWMaleData$years, ages.fit = 55:89)
plot(M7fit, parametricbx = FALSE)
```

m8	<i>Create an M8 type extension of the Cairns-Blake-Dowd mortality model</i>
----	---

Description

Utility function to initialise a StMoMo object representing the M8 extension of the Cairns-Blake-Dowd mortality model introduced in Cairns et al (2009).

Usage

```
m8(link = c("logit", "log"), xc)
```

Arguments

link	defines the link function and random component associated with the mortality model. "log" would assume that deaths follow a Poisson distribution and use a log link while "logit" would assume that deaths follow a Binomial distribution and a logit link. Note that the default is the logit link.
xc	constant defining the cohort age-modulating parameter.

Details

The created model is either a logit-Binomial or a log-Poisson version of the M8 model which has predictor structure

$$\eta_{xt} = \kappa_t^{(1)} + (x - \bar{x})\kappa_t^{(2)} + (x_c - x)\gamma_{t-x}$$

where \bar{x} is the average age in the data and x_c is a predefined constant. Identifiability of the model is accomplished by applying parameters constraint

$$\sum_c \gamma_c = 0.$$

Value

An object of class "StMoMo".

References

Cairns, A. J. G., Blake, D., Dowd, K., Coughlan, G. D., Epstein, D., Ong, A., & Balevich, I. (2009). A quantitative comparison of stochastic mortality models using data from England and Wales and the United States. *North American Actuarial Journal*, 13(1), 1-35.

See Also

[StMoMo](#), [cbd](#), [m6](#), [m7](#)

Examples

```
M8 <- m8(xc = 89)
Dxt <- EWMaleData$Dxt
Ext <- EWMaleData$Ext + 0.5 * EWMaleData$Dxt
wxt <- genWeightMat(55:89, EWMaleData$years, clip = 3)
M8fit <- fit(M8, Dxt = Dxt, Ext = Ext, ages = EWMaleData$ages,
             years = EWMaleData$years, ages.fit = 55:89)
plot(M8fit, parametricbx = FALSE)
```

Description

Fits a Multivariate Random Walk with Drift to x , a multivariate time series.

Usage

```
mrwd(x)
```

Arguments

`x` numeric matrix with a multivariate time series. Series are arranged in rows with columns representing time.

Details

For further information on the Multivariate Random Walk with drift see Appendix B in Haberman and Renshaw (2011).

Value

an object of class "mrwd" with components:

<code>drift</code>	a vector with the estimated drift.
<code>sigma</code>	a matrix with the estimated variance covariance matrix.
<code>fitted</code>	fitted values.
<code>residuals</code>	residuals from the fitted model. That is observed minus fitted values.
<code>x</code>	the original time series.

References

Haberman, S., & Renshaw, A. (2011). A comparative study of parametric mortality projection models. *Insurance: Mathematics and Economics*, 48(1), 35-55.

plot.bootStMoMo	<i>Plot bootstrapped parameters of a Stochastic Mortality Model</i>
-----------------	---

Description

Plot fancharts of bootstrapped parameters of a Stochastic Mortality Model stored in an object of class "bootStMoMo".

Usage

```
## S3 method for class 'bootStMoMo'
plot(x, nCol = 2, parametricbx = TRUE,
      colour = rgb(0, 0, 0),
      probs = c(2.5, 10, 25, 50, 75, 90, 97.5), ...)
```

Arguments

<code>x</code>	an object of class "bootStMoMo" with the bootstrapped parameters of a stochastic mortality model.
<code>nCol</code>	number of columns to use in the plot.
<code>parametricbx</code>	if FALSE parametric age-modulating terms, which don't need to be estimated, are not plotted.

colour	colour to use in the fans.
probs	probabilities related to percentiles to plot in the fan chart. The default <code>c(2.5, 10, 25, 50, 75, 90, 97.5)</code> plots the 50%, 80% and 95% confidence intervals of the parameters.
...	other arguments.

See Also

[plot.fitStMoMo](#)

Examples

```
#Long computing times
## Not run:
CBDfit <- fit(cbd(),Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
             ages = EWMaleData$ages, years = EWMaleData$years,
             ages.fit = 55:89)
CBDResBoot <- bootstrap(CBDfit, nBoot = 500)
plot(CBDResBoot)
plot(CBDResBoot, parametricbx = FALSE, probs = seq(2.5, 97.5, 2.5))

## End(Not run)
```

plot.fitStMoMo	<i>Plot fitted parameters from a stochastic mortality model</i>
----------------	---

Description

Plot fitted parameters of a stochastic mortality model of class "fitStMoMo".

Usage

```
## S3 method for class 'fitStMoMo'
plot(x, nCol = 2, parametricbx = TRUE, type = "l", ...)
```

Arguments

x	an object of class "fitStMoMo" with the fitted parameters of a stochastic mortality model.
nCol	number of columns to use in the plot.
parametricbx	if FALSE parametric age-modulating terms, which don't need to be estimated, are not plotted.
type	what type of plot should be drawn. See plot .
...	additional arguments to control graphical appearance. See plot .

Examples

```
#Fit and plot a Lee-Carter model
LCfit <- fit(lc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
            ages = EWMaleData$ages, years = EWMaleData$years,
            ages.fit = 55:89)
plot(LCfit)
plot(LCfit, type = "p", pch = 19)

#Fit and plot a CBD model
CBDfit <- fit(cbd(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
            ages = EWMaleData$ages, years = EWMaleData$years,
            ages.fit = 55:89)
plot(CBDfit)
plot(CBDfit, parametricbx = FALSE)
plot(CBDfit, nCol = 1, parametricbx = FALSE, lwd = 2)
```

plot.forStMoMo	<i>Plot a forecast from a Stochastic Mortality Model</i>
----------------	--

Description

Plot a forecasted Stochastic Mortality Model of class "forStMoMo".

Usage

```
## S3 method for class 'forStMoMo'
plot(x, nCol = 2, parametricbx = TRUE, only.kt = FALSE,
     only.gc = FALSE, ...)
```

Arguments

x	an object of class "forStMoMo" with the forecast of a stochastic mortality model.
nCol	number of columns to use in the plot.
parametricbx	if FALSE parametric age-modulating terms, which don't need to be estimated, are not plotted.
only.kt	If TRUE only the period indexes of the model are plotted.
only.gc	If TRUE only the cohort index of the model is plotted. This argument is ignored if only.kt is TRUE.
...	additional arguments to control graphical appearance. See plot .

See Also

[plot.fitStMoMo](#)

Examples

```
wxt <- genWeightMat(55:89, EWMaleData$years, clip = 3)
APCfit <- fit(apc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
             ages = EWMaleData$ages, years = EWMaleData$years,
             ages.fit = 55:89, wxt = wxt)
APCfor <- forecast(APCfit)
plot(APCfor)
plot(APCfor, parametricbx = FALSE, nCol = 3)
plot(APCfor, only.kt = TRUE)
plot(APCfor, only.gc = TRUE, lwd = 2)
```

plot.resStMoMo

Plot the residuals of a Stochastic Mortality Model

Description

Plots the deviance residuals of a Stochastic Mortality Model which are of class "resStMoMo". Three types of plots are available: scatter plot of residuals by age, period and cohort, colour map (heatmap) of the residuals, and a black and white signplot of the residuals.

Usage

```
## S3 method for class 'resStMoMo'
plot(x, type = c("scatter", "colourmap",
                "signplot"),
     reslim = NULL, plotAge = TRUE,
     plotYear = TRUE, plotCohort = TRUE,
     pch = 20, col = NULL, ...)
```

Arguments

x	an object of class resStMoMo with the residuals of a Stochastic Mortality Model.
type	the type of the plot. The alternatives are "scatter" (default), "colourmap", and "signplot".
reslim	optional numeric vector of length 2, giving the range of the residuals.
plotAge	logical value indicating if the age scatter plot should be produced. This is only used when type = "scatter".
plotYear	logical value indicating if the calendar year scatter plot should be produced. This is only used when type = "scatter".
plotCohort	logical value indicating if the cohort scatter plot should be produced. This is only used when type = "scatter".
pch	optional symbol to use for the points in a scatterplot. This is only used when type = "scatter". See plot .

`col` optional colours to use in plotting. If `type = "scatter"` this is a single colour to use in the points in the scatter plots, while if `type = "colourmap"` this should be a list of colours (see help in [image.plot](#) for details). This argument is ignored if `type = "signplot"`.

`...` other plotting parameters to be passed to the plotting functions. This can be used to control the appearance of the plots.

Details

When `type = "scatter"` scatter plots of the residuals against age, calendar year and cohort (year of birth) are produced.

When `type = "colourmap"` a two dimensional colour map of the residuals is plotted. This is produced using function [image.plot](#). See [image.plot](#) for further parameters that can be passed to this type of plots.

When `type = "signplot"` a two dimensional black and white map of the residuals is plotted with dark grey representing negative residuals and light grey representing positive residuals. This is produced using function [image.default](#).

@seealso [residuals.fitStMoMo](#)

Examples

```
CBDfit <- fit(cbd(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
             ages = EWMaleData$ages, years = EWMaleData$years,
             ages.fit = 55:89)
CBDres <- residuals(CBDfit)
plot(CBDres)
plot(CBDres, type = "signplot")
plot(CBDres, type = "colourmap")
```

predict.fitStMoMo

Predict method for Stochastic Mortality Models fits

Description

Obtain predictions from a Stochastic Mortality Model fit.

Usage

```
## S3 method for class 'fitStMoMo'
predict(object, years, kt = NULL, gc = NULL,
        oxt = NULL, type = c("link", "rates"), ...)
```

Arguments

object	an object of class "fitStMoMo" with the fitted parameters of a stochastic mortality model.
years	vector of years for which a prediction is required.
kt	matrix of values of the period indexes to use for the prediction. If the model has any age-period term this argument needs to be provided and the number of rows in kt must be equal to the number of age-period terms in the model and the number of columns in kt must correspond to the length of years. If the Stochastic Mortality Model doesn't have any age-period terms this argument is ignored and needs not be provided.
gc	vector of values of the cohort indexes to use for the prediction. If the model has a cohort effect this argument needs to be provided. In this case the length of gc must be equal to the number of cohorts for which a prediction is being produced, namely, $\text{length}(\text{object}\$ages) + \text{length}(\text{years}) - 1$. If the Stochastic Mortality Model doesn't have a cohort effect this argument is ignored and needs not be provided.
oxt	optional matrix/vector or scalar of known offset to be used in the prediction.
type	the type of the predicted values that should be returned. The alternatives are "link"(default) and "rates".
...	other arguments.

Details

This function evaluates

$$\hat{\eta}_{xt} = o_{xt} + \alpha_x + \sum_{i=1}^N \beta_x^{(i)} \kappa_t^{(i)} + \beta_x^{(0)} \gamma_{t-x}$$

for a fitted Stochastic Mortality model. In producing a prediction the static age function, α_x , and the age-modulating parameters, $\beta_x^{(i)}$, $i = 0, \dots, N$, are taken from the fitted model in object while the period indexes, $\kappa_t^{(i)}$, $i = 1, \dots, N$, and cohort index, γ_{t-x} , are taken from the function arguments.

This function can be useful, for instance, in producing forecasts of mortality rates using time series models different to those available in [forecast.fitStMoMo](#) (See examples below).

Value

A matrix with the predicted values.

See Also

[forecast.fitStMoMo](#)

Examples

```
library(forecast)
#Lee-Carter forecast using auto.arima
LCfit <- fit(lc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
```

```

      ages = EWMaleData$ages, years = EWMaleData$years,
      ages.fit = 55:89)
ktForLC <- forecast(auto.arima(as.vector(LCfit$kt)), h = 30)
mxtForLC <- predict(LCfit, years = 2012:2041, kt = ktForLC$mean,
                    type = "rates")
mxthatLC <- fitted(LCfit, type = "rates")
mxt <- LCfit$Dxt / LCfit$Ext
plot(1961:2041, (cbind(mxthatLC, mxtForLC))["80", ], type = "l",
     xlab = "year", ylab = "death rate",
     main = "Fitted vs. Observed rates at age 80")
points(1961:2011, mxt["80", ])

#Age-Period-Cohort forecast using auto.arima
APCfit <- fit(apc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
             ages = EWMaleData$ages, years = EWMaleData$years,
             ages.fit = 55:89)
ktForAPC <- forecast(auto.arima(as.vector(APCfit$kt)), h = 30)
gcForAPC <- forecast(auto.arima(as.vector(APCfit$gc), max.d = 1), h = 30)
mxtForAPC <- predict(APCfit, years = 2012:2041, kt = ktForAPC$mean,
                    gc = c(tail(APCfit$gc, 34), gcForAPC$mean),
                    type = "rates")
mxthatAPC <- fitted(APCfit, type = "rates")
lines(1961:2041 , (cbind(mxthatAPC, mxtForAPC))["80", ], type = "l",
     col = "blue")

```

print.bootStMoMo

Print an object of class "bootStMoMo"

Description

print method for class "bootStMoMo".

Usage

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

Arguments

x an object of class "bootStMoMo".

... arguments to be passed to or from other methods.

print.fitStMoMo	<i>Print an object of class "fitStMoMo"</i>
-----------------	---

Description

print method for class "fitStMoMo".

Usage

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

Arguments

x	an object of class "fitStMoMo".
...	arguments to be passed to or from other methods.

print.forStMoMo	<i>Print an object of class "forStMoMo"</i>
-----------------	---

Description

print method for class "forStMoMo".

Usage

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

Arguments

x	an object of class "forStMoMo".
...	arguments to be passed to or from other methods.

<code>print.simStMoMo</code>	<i>Print an object of class "simStMoMo"</i>
------------------------------	---

Description

print method for class "simStMoMo".

Usage

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

Arguments

- x an object of class "simStMoMo".
- ... arguments to be passed to or from other methods.

<code>print.StMoMo</code>	<i>Print an object of class "StMoMo"</i>
---------------------------	--

Description

print method for class "StMoMo".

Usage

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

Arguments

- x an object of class "StMoMo".
- ... arguments to be passed to or from other methods.

Details

`print.StMoMo` prints a description of the Stochastic Mortality Model

residuals.fitStMoMo	<i>Extract deviance residuals of a Stochastic Mortality Model</i>
---------------------	---

Description

Compute deviance residuals of a fitted Stochastic Mortality Model. These residuals can be plotted using [plot.resStMoMo](#).

Usage

```
## S3 method for class 'fitStMoMo'
residuals(object, scale = TRUE, ...)
```

Arguments

object	an object of class "fitStMoMo" with the fitted parameters of a stochastic mortality model.
scale	logical indicating whether the residuals should be scaled or not by dividing the deviance by the overdispersion of the model. Default is TRUE.
...	other arguments.

Value

An object of class "resStMoMo" with the residuals. This object has components:

residuals	a matrix with the residuals.
ages	ages corresponding to the rows in residuals.
years	years corresponding to the columns in residuals.

See Also

[plot.resStMoMo](#)

Examples

```
CBDfit <- fit(cbd(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
             ages = EWMaleData$ages, years = EWMaleData$years,
             ages.fit = 55:89)
CBDres <- residuals(CBDfit)
plot(CBDres)
```

rh	<i>Create a Renshaw and Haberman (Lee-Carter with cohorts) mortality model</i>
----	--

Description

Utility function to initialise a StMoMo object representing a Renshaw and Haberman (Lee-Carter with cohorts) mortality model introduced in Renshaw and Haberman (2006).

Usage

```
rh(link = c("log", "logit"), cohortAgeFun = c("1", "NP"))
```

Arguments

link	defines the link function and random component associated with the mortality model. "log" would assume that deaths follow a Poisson distribution and use a log link while "logit" would assume that deaths follow a Binomial distribution and a logit link.
cohortAgeFun	defines the cohort age modulating parameter $\beta_x^{(0)}$. It can take values: "NP" for a non-parametric age term or "1" for $\beta_x^{(0)} = 1$ (the default).

Details

The created model is either a log-Poisson or a logit-Binomial version of the Renshaw and Haberman model which has predictor structure

$$\eta_{xt} = \alpha_x + \beta_x^{(1)}\kappa_t + \beta^{(0)}\gamma_{t-x}.$$

or

$$\eta_{xt} = \alpha_x + \beta_x^{(1)}\kappa_t + \gamma_{t-x}.$$

depending on the value of argument cohortAgeFun.

To ensure identifiability the following constraints are imposed

$$\sum_t \kappa_t = 0, \sum_x \beta_x^{(1)} = 1, \sum_c \gamma_c = 0$$

plus

$$\sum_x \beta_x^{(0)} = 1$$

if cohortAgeFun = "NP"

By default $\beta_x^{(0)} = 1$ as this model has shown to be more stable (see Haberman and Renshaw (2011) and Hunt and Villegas (2015)).

Value

An object of class "StMoMo".

References

- Haberman, S., & Renshaw, A. (2011). A comparative study of parametric mortality projection models. *Insurance: Mathematics and Economics*, 48(1), 35-55.
- Hunt, A., & Villegas, A. M. (2015). Robustness and convergence in the Lee-Carter model with cohorts. *Insurance: Mathematics and Economics*, 64, 186-202.
- Renshaw, A. E., & Haberman, S. (2006). A cohort-based extension to the Lee-Carter model for mortality reduction factors. *Insurance: Mathematics and Economics*, 38(3), 556-570.

See Also

[StMoMo](#), [lc](#), [apc](#)

Examples

```
LCfit <- fit(lc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
             ages = EWMaleData$ages, years = EWMaleData$years,
             ages.fit = 55:89)
wxt <- genWeightMat(55:89, EWMaleData$years, clip = 3)
RHfit <- fit(rh(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
             ages = EWMaleData$ages, years = EWMaleData$years,
             ages.fit = 55:89, wxt = wxt, start.ax = LCfit$ax,
             start.bx = LCfit$bx, start.kt = LCfit$kt)
plot(RHfit)
```

simulate.bootStMoMo	<i>Simulate future sample paths from a Bootstrapped Stochastic Mortality Model</i>
---------------------	--

Description

Simulate future sample paths from a Bootstrapped Stochastic Mortality Model. The period indexes $\kappa_t^{(i)}$, $i = 1, \dots, N$, are modelled using a Multivariate Random Walk with Drift. The cohort index γ_{t-x} is modelled using an ARIMA(p, d, q). By default an ARIMA(1, 1, 0) with a constant is used.

Usage

```
## S3 method for class 'bootStMoMo'
simulate(object, nsim = 1, seed = NULL, h = 50,
         oxt = NULL, gc.order = c(1, 1, 0), gc.include.constant = TRUE,
         jumpchoice = c("fit", "actual"), kt.lookback = NULL, gc.lookback = NULL,
         ...)
```

Arguments

object	an object of class "bootStMoMo" with the bootstrapped parameters of a stochastic mortality model.
nsim	number of sample paths to simulate from each bootstrapped sample. Thus if there are nBoot bootstrapped samples the total number of paths will be nsim * nBoot.
seed	either NULL or an integer that will be used in a call to set.seed before simulating the time series. The default, NULL will not change the random generator state.
h	number of years ahead to forecast.
oxt	optional array/matrix/vector or scalar of known offset to be added in the simulations. This can be used to specify any a priori known component to be added to the simulated predictor.
gc.order	a specification of the ARIMA model: the three components (p, d, q) are the AR order, the degree of differencing, and the MA order. The default is an ARIMA(1, 1, 0).
gc.include.constant	a logical value indicating if the ARIMA model should include a constant value. The default is TRUE.
jumpchoice	option to select the jump-off rates, i.e. the rates from the final year of observation, to use in projections of mortality rates. "fit"(default) uses the fitted rates and "actual" uses the actual rates from the final year.
kt.lookback	optional argument to specify the look-back window to use in the estimation of the MRWD for period indexes. By default all the estimated values are used in estimating the MRWD. If kt.lookback is provided then the last kt.lookback years of $\kappa_t^{(i)}, i = 1, \dots, N$, are used.
gc.lookback	optional argument to specify the look-back window to use in the estimation of the ARIMA model for the cohort effect. By default all the estimated values are used in estimating the ARIMA model. If gc.lookback is provided then the last gc.lookback years of γ_{t-x} are used.
...	other arguments.

Details

For further details see [simulate.fitStMoMo](#).

Value

A list of class "simStMoMo" with components

rates	a three dimensional array with the future simulated rates.
ages	vector of ages corresponding to the first dimension of rates.
years	vector of years for which a simulations has been produced. This corresponds to the second dimension of rates.
kt.s	information on the simulated paths of the period indices of the model. This is a list with the simulated paths of κ_t (sim) and the years for which simulations were produced. If the mortality model does not have any age-period terms (i.e. $N = 0$) this is set to NULL.

gc.s	information on the simulated paths of the cohort index of the model. This is a list with the simulated paths of γ_c (sim) and the cohorts for which simulations were produced. If the mortality model does not have a cohort effect this is set to NULL.
oxt.s	a three dimensional array with the offset used in the simulations.
fitted	a three dimensional array with the in-sample rates of the model for the years for which the mortality model was fitted (and bootstrapped).
jumpchoice	Jump-off method used in the simulation.
model	the bootstrapped model from which the simulations were produced.

See Also

[bootstrap.fitStMoMo](#), [simulate.fitStMoMo](#)

Examples

```
#Long computing times
## Not run:
#Lee-Carter: Compare projection with and without parameter uncertainty
library(fanplot)
LCfit <- fit(lc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
            ages = EWMaleData$ages, years = EWMaleData$years)
LCResBoot <- bootstrap(LCfit, nBoot = 500)
LCResBootsim <- simulate(LCResBoot)
LCsim <- simulate(LCfit, nsim = 500) #'
plot(LCfit$years, log(LCfit$Dxt / LCfit$Ext)["10", ],
     xlim = range(LCfit$years, LCsim$years),
     ylim = range(log(LCfit$Dxt / LCfit$Ext)["10", ],
                  log(LCsim$rates["10", , ])),
     type = "l", xlab = "year", ylab = "log rate",
     main = "Mortality rate projection at age 10 with and without parameter uncertainty")
fan(t(log(LCResBootsim$rates["10", , ])), start = LCResBootsim$years[1],
    probs = c(2.5, 10, 25, 50, 75, 90, 97.5), n.fan = 4,
    fan.col = colorRampPalette(c(rgb(0, 0, 1), rgb(1, 1, 1))), ln = NULL)
fan(t(log(LCsim$rates["10", 1:(length(LCsim$years) - 3), ])),
    start = LCsim$years[1], probs = c(2.5, 10, 25, 50, 75, 90, 97.5),
    n.fan = 4, fan.col = colorRampPalette(c(rgb(1, 0, 0), rgb(1, 1, 1))),
    ln = NULL)

## End(Not run)
```

Description

Simulate future sample paths from a Stochastic Mortality Model. The period indexes $\kappa_t^{(i)}, i = 1, \dots, N$, are modelled using a Multivariate Random Walk with Drift. The cohort index γ_{t-x} is modelled using an ARIMA(p, d, q). By default an ARIMA(1, 1, 0) with a constant is used.

Usage

```
## S3 method for class 'fitStMoMo'
simulate(object, nsim = 1000, seed = NULL, h = 50,
  oxt = NULL, gc.order = c(1, 1, 0), gc.include.constant = TRUE,
  jumpchoice = c("fit", "actual"), kt.lookback = NULL, gc.lookback = NULL,
  ...)
```

Arguments

object	an object of class "fitStMoMo" with the fitted parameters of a stochastic mortality model.
nsim	number of sample paths to simulate.
seed	either NULL or an integer that will be used in a call to set.seed before simulating the time series. The default, NULL will not change the random generator state.
h	number of years ahead to forecast.
oxt	optional matrix/vector or scalar of known offset to be added in the simulations. This can be used to specify any a priori known component to be added to the simulated predictor.
gc.order	a specification of the ARIMA model: the three components (p, d, q) are the AR order, the degree of differencing, and the MA order. The default is an ARIMA(1, 1, 0).
gc.include.constant	a logical value indicating if the ARIMA model should include a constant value. The default is TRUE.
jumpchoice	option to select the jump-off rates, i.e. the rates from the final year of observation, to use in projections of mortality rates. "fit"(default) uses the fitted rates and "actual" uses the actual rates from the final year.
kt.lookback	optional argument to specify the look-back window to use in the estimation of the MRWD for period indexes. By default all the estimated values are used in estimating the MRWD. If kt.lookback is provided then the last kt.lookback years of $\kappa_t^{(i)}, i = 1, \dots, N$, are used.
gc.lookback	optional argument to specify the look-back window to use in the estimation of the ARIMA model for the cohort effect. By default all the estimated values are used in estimating the ARIMA model. If gc.lookback is provided then the last gc.lookback years of γ_{t-x} are used.
...	other arguments.

Details

Fitting and simulation of the Multivariate Random Walk with Drift for the period indexes is done using the function `mrwd`. Fitting and simulation of the ARIMA model for the cohort index is done with function `Arima` from package **forecast**. See the latter function for further details on input arguments `gc.order` and `gc.include.constant`.

Note that in some cases simulations of the cohort effects may be needed for a horizon longer than `h`. This is the case when in the fitted model the most recent cohorts have been zero weighted. The simulated cohorts can be seen in `gc.s$cohorts`.

Value

A list of class "simStMoMo" with components:

<code>rates</code>	a three dimensional array with the future simulated rates.
<code>ages</code>	vector of ages corresponding to the first dimension of <code>rates</code> .
<code>years</code>	vector of years for which a simulations has been produced. This corresponds to the second dimension of <code>rates</code> .
<code>kt.s</code>	information on the simulated paths of the period indexes of the model. This is a list with the model fitted to κ_t ; the simulated paths (<code>sim</code>); and the years for which simulations were produced. If the mortality model does not have any age-period terms (i.e. $N = 0$) this is set to <code>NULL</code> .
<code>gc.s</code>	information on the simulated paths of the cohort index of the model. This is a list with the model fitted to γ_c ; the simulated paths (<code>sim</code>); and the cohorts for which simulations were produced. If the mortality model does not have a cohort effect this is set to <code>NULL</code> .
<code>oxt.s</code>	a three dimensional array with the offset used in the simulations.
<code>fitted</code>	a three dimensional array with the in-sample rates of the model for the years for which the mortality model was fitted.
<code>jumpchoice</code>	Jump-off method used in the simulation.
<code>model</code>	the model fit from which the simulations were produced.

See Also

[forecast.fitStMoMo](#)

Examples

```
#Lee-Carter
LCfit <- fit(lc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
            ages = EWMaleData$ages, years = EWMaleData$years,
            ages.fit = 55:89)
LCsim <- simulate(LCfit, nsim = 100)

par(mfrow=c(1, 2))
plot(LCfit$years, LCfit$kt[1, ], xlim = range(LCfit$years, LCsim$kt.s$years),
     ylim = range(LCfit$kt, LCsim$kt.s$sim), type = "l",
     xlab = "year", ylab = "kt",
```

```

    main = "Lee-Carter: Simulated paths of the period index kt")
matlines(LCsim$kt.s$years, LCsim$kt.s$sim[1, , ], type = "l", lty = 1)

plot(LCfit$years, (LCfit$Dxt / LCfit$Ext)["65", ],
     xlim = range(LCfit$years, LCsim$years),
     ylim = range((LCfit$Dxt / LCfit$Ext)["65", ], LCsim$rates["65", , ]),
     type = "l", xlab = "year", ylab = "rate",
     main = "Lee-Carter: Simulated mortality rates at age 65")
matlines(LCsim$years, LCsim$rates["65", , ], type = "l", lty = 1)

#APC
par(mfrow=c(1, 3))
wxt <- genWeightMat(55:89, EWMaleData$years, clip = 3)
APCfit <- fit(apc(), Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
             ages = EWMaleData$ages, years = EWMaleData$years,
             ages.fit = 55:89, wxt = wxt)
APCsim <- simulate(APCfit, nsim = 100, gc.order = c(1, 1, 0))

plot(APCfit$years, APCfit$kt[1, ],
     xlim = range(APCfit$years, APCsim$kt.s$years),
     ylim = range(APCfit$kt, APCsim$kt.s$sim), type = "l",
     xlab = "year", ylab = "kt",
     main = "APC: Simulated paths of the period index kt")
matlines(APCsim$kt.s$years, APCsim$kt.s$sim[1, , ], type = "l", lty = 1)

plot(APCfit$cohorts, APCfit$gc,
     xlim = range(APCfit$cohorts, APCsim$gc.s$cohorts),
     ylim = range(APCfit$gc, APCsim$gc.s$sim, na.rm = TRUE), type = "l",
     xlab = "year", ylab = "kt",
     main = "APC: Simulated paths of the cohort index (ARIMA(1,1,0))")
matlines(APCsim$gc.s$cohorts, APCsim$gc.s$sim, type = "l", lty = 1)

plot(APCfit$years, (APCfit$Dxt / APCfit$Ext)["65", ],
     xlim = range(APCfit$years, APCsim$years),
     ylim = range((APCfit$Dxt / APCfit$Ext)["65", ], APCsim$rates["65", , ]),
     type = "l", xlab = "year", ylab = "rate",
     main = "APC: Simulated of mortality rates at age 65")
matlines(APCsim$years, APCsim$rates["65", , ], type = "l", lty = 1)

#Compare LC and APC
library(fanplot)
par(mfrow=c(1, 1))
plot(LCfit$years, (LCfit$Dxt / LCfit$Ext)["65", ],
     xlim = range(LCfit$years, LCsim$years),
     ylim = range((LCfit$Dxt / LCfit$Ext)["65", ], LCsim$rates["65", , ],
     APCsim$rates["65", , ], type = "l", xlab = "year", ylab = "rate",
     main = "Fan chart of mortality rates at age 65 (LC vs. APC)")
fan(t(LCsim$rates["65", , ]), start = LCsim$years[1],
    probs = c(2.5, 10, 25, 50, 75, 90, 97.5), n.fan = 4,
    fan.col = colorRampPalette(c(rgb(1, 0, 0), rgb(1, 1, 1))), ln = NULL)
fan(t(APCsim$rates["65", 1:(length(APCsim$years) - 3), ]),
    start = APCsim$years[1], probs = c(2.5, 10, 25, 50, 75, 90, 97.5),
    n.fan = 4, fan.col = colorRampPalette(c(rgb(0, 0, 1), rgb(1, 1, 1))),

```



```
ln = NULL)
```

simulate.mrwd	<i>Simulate a Multivariate Random Walk with Drift</i>
---------------	---

Description

Returns one simulated path of the Multivariate Random Walk with Drift model in object.

Usage

```
## S3 method for class 'mrwd'
simulate(object, nsim = 10, seed = NULL, ...)
```

Arguments

- | | |
|--------|--|
| object | An object of class "mrwd". |
| nsim | number of periods for the simulated series. |
| seed | either NULL or an integer that will be used in a call to set.seed before simulating the time series. The default, NULL will not change the random generator state. |
| ... | other arguments. |

StMoMo	<i>Create a new Stochastic Mortality Model</i>
--------	--

Description

Initialises a StMoMo object which represents a Generalised Age-Period-Cohort Stochastic Mortality Model.

StMoMo.

Usage

```
StMoMo(link = c("log", "logit"), staticAgeFun = TRUE, periodAgeFun = "NP",
        cohortAgeFun = NULL, constFun = function(ax, bx, kt, b0x, gc, wxt, ages)
        list(ax = ax, bx = bx, kt = kt, b0x = b0x, gc = gc))
```

Arguments

link	defines the link function and random component associated with the mortality model. "log" would assume that deaths follow a Poisson distribution and use a log link while "logit" would assume that deaths follow a Binomial distribution and a logit link.
staticAgeFun	logical value indicating if a static age function α_x is to be included.
periodAgeFun	a list of length N with the definitions of the period age modulating parameters $\beta_x^{(i)}$. Each entry can take values: "NP" for non-parametric age terms, "1" for $\beta_x^{(i)} = 1$ or a predefined parametric function of age (see details). Set this to NULL if there are no period terms in the model.
cohortAgeFun	defines the cohort age modulating parameter $\beta_x^{(0)}$. It can take values: "NP" for non-parametric age terms, "1" for $\beta_x^{(0)} = 1$, a predefined parametric function of age (see details) or NULL if there is no cohort effect.
constFun	function defining the identifiability constraints of the model. It must be a function of the form <code>constFun <- function(ax, bx, kt, b0x, gc, wxt, ages)</code> taking a set of fitted model parameters and returning a list <code>list(ax = ax, bx = bx, kt = kt, b0x = b0x, gc = gc, wxt = wxt)</code> of the model parameters with the identifiability constraints applied. If omitted no identifiability constraints are applied to the model.

Details

R implementation of the family of Generalised Age-Period-Cohort stochastic mortality models. This family of models encompasses many models proposed in the literature including the well-known Lee-Carter model, CBD model and APC model.

StMoMo defines an abstract representation of a Generalised Age-Period-Cohort (GAPC) Stochastic model that fits within the general class of generalised non-linear models defined as follows

$$D_{xt} \sim \text{Poisson}(E_{xt}\mu_{xt}), D_{xt} \sim \text{Binomial}(E_{xt}, q_{xt})$$

$$\eta_{xt} = \log \mu_{xt}, \eta_{xt} = \text{logit } q_{xt}$$

$$\eta_{xt} = \alpha_x + \sum_{i=1}^N \beta_x^{(i)} \kappa_t^{(i)} + \beta_x^{(0)} \gamma_{t-x}$$

$$v : \{\alpha_x, \beta_x^{(1)}, \dots, \beta_x^{(N)}, \kappa_t^{(1)}, \dots, \kappa_t^{(N)}, \beta_x^{(0)}, \gamma_{t-x}\} \mapsto \{\alpha_x, \beta_x^{(1)}, \dots, \beta_x^{(N)}, \kappa_t^{(1)}, \dots, \kappa_t^{(N)}, \beta_x^{(0)}, \gamma_{t-x}\},$$

where

- α_x is a static age function;
- $\beta_x^{(i)} \kappa_t^{(i)}, i = 1, \dots, N$, are age/period terms;
- $\beta_x^{(0)} \gamma_{t-x}$ is the age/cohort term; and
- v is a function defining the identifiability constraints of the model.

Most Stochastic mortality models proposed in the literature can be cast to this representation (See Hunt and Blake (2015)).

Parametric age functions should be scalar functions of the form `f <- function(x, ages)` taking a scalar age x and a vector of model fitting ages (see examples below).

Do to limitation of functions `gnm` within package `gnm`, which is used for fitting "StMoMo" objects to data (see `fit.StMoMo`), models combining parametric and non-parametric age-modulating functions are not supported at the moment.

Value

A list with class "StMoMo" with components:

<code>link</code>	a character string defining the link function of the model.
<code>staticAgeFun</code>	a logical value indicating if the model has a static age function.
<code>periodAgeFun</code>	a list defining the period age modulating parameters.
<code>cohortAgeFun</code>	an object defining the cohort age modulating parameters.
<code>constFun</code>	a function defining the identifiability constraints.
<code>N</code>	an integer specifying The number of age-period terms in the model.
<code>textFormula</code>	a character string of the model formula.
<code>gnmFormula</code>	a formula that can be used for fitting the model with package <code>gnm</code> .

References

- Plat, R. (2009). On stochastic mortality modeling. *Insurance: Mathematics and Economics*, 45(3), 393-404.
- Hunt, A., & Blake, D. (2015). On the Structure and Classification of Mortality Models Mortality Models. Pension Institute Working Paper. <http://www.pensions-institute.org/workingpapers/wp1506.pdf>.

See Also

`fit.StMoMo`, `lc`, `cbd`, `apc`, `rh`, `m6`, `m7`, `m8`

Examples

```
#Lee-Carter model
constLC <- function(ax, bx, kt, b0x, gc, wxt, ages) {
  c1 <- mean(kt[1, ], na.rm = TRUE)
  c2 <- sum(bx[, 1], na.rm = TRUE)
  list(ax = ax + c1 * bx, bx = bx / c2, kt = c2 * (kt - c1))
}
LC <- StMoMo(link = "log", staticAgeFun = TRUE, periodAgeFun = "NP",
             constFun = constLC)

plot(fit(LC, Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
         ages = EWMaleData$ages, years = EWMaleData$years,
         ages.fit = 55:89))

#CBD model
f2 <- function(x, ages) x - mean(ages)
CBD <- StMoMo(link = "logit", staticAgeFun = FALSE,
              periodAgeFun = c("1", f2))
```

```

plot(fit(CBD, Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
        ages = EWMaleData$ages, years = EWMaleData$years,
        ages.fit = 55:89))

#Reduced Plat model (Plat, 2009)
f2 <- function(x, ages) mean(ages) - x
constPlat <- function(ax, bx, kt, b0x, gc, wxt, ages) {
  nYears <- dim(wxt)[2]
  x <- ages
  t <- 1:nYears
  c <- (1 - tail(ages, 1)):(nYears - ages[1])
  xbar <- mean(x)
  #nsum g(c)=0, nsum cg(c)=0, nsum c^2g(c)=0
  phiReg <- lm(gc ~ 1 + c + I(c^2), na.action = na.omit)
  phi <- coef(phiReg)
  gc <- gc - phi[1] - phi[2] * c - phi[3] * c^2
  kt[2, ] <- kt[2, ] + 2 * phi[3] * t
  kt[1, ] <- kt[1, ] + phi[2] * t + phi[3] * (t^2 - 2 * xbar * t)
  ax <- ax + phi[1] - phi[2] * x + phi[3] * x^2
  #nsum kt[i, ] = 0
  ci <- rowMeans(kt, na.rm = TRUE)
  ax <- ax + ci[1] + ci[2] * (xbar - x)
  kt[1, ] <- kt[1, ] - ci[1]
  kt[2, ] <- kt[2, ] - ci[2]
  list(ax = ax, bx = bx, kt = kt, b0x = b0x, gc = gc)
}
PLAT <- StMoMo(link = "log", staticAgeFun = TRUE,
              periodAgeFun = c("1", f2), cohortAgeFun = "1",
              constFun = constPlat)

plot(fit(PLAT, Dxt = EWMaleData$Dxt, Ext = EWMaleData$Ext,
        ages = EWMaleData$ages, years = EWMaleData$years,
        ages.fit = 55:89))

#Models not supported
## Not run:
MnotSup1 <- StMoMo(periodAgeFun = c(f2, "NP"))
MnotSup1 <- StMoMo(periodAgeFun = f2, cohortAgeFun = "NP")

## End(Not run)

```

Index

*Topic **datasets**

EWMaleData, 8

apc, 2, 35, 43

Arima, 14, 39

bootstrap, 4

bootstrap.fitStMoMo, 4, 4, 37

cbd, 6, 21–23, 43

coef.fitStMoMo, 7

EWMaleData, 8

fit, 9

fit.StMoMo, 9, 9, 17, 43

fitted.fitStMoMo, 12

forecast.fitStMoMo, 13, 29, 39

forecast.mrwd, 16

genWeightMat, 10, 11, 17

gnm, 10, 43

image.default, 28

image.plot, 28

lc, 18, 35, 43

logLik.fitStMoMo, 19

m6, 6, 20, 22, 23, 43

m7, 6, 21, 21, 23, 43

m8, 6, 21, 22, 22, 43

mrwd, 14, 23, 39

plot, 25–27

plot.bootStMoMo, 5, 24

plot.fitStMoMo, 10, 11, 25, 25, 26

plot.forStMoMo, 26

plot.resStMoMo, 27, 33

predict.fitStMoMo, 28

print.bootStMoMo, 30

print.fitStMoMo, 31

print.forStMoMo, 31

print.simStMoMo, 32

print.StMoMo, 32

residuals.fitStMoMo, 28, 33

rh, 3, 34, 43

set.seed, 36, 38, 41

simulate.bootStMoMo, 5, 35

simulate.fitStMoMo, 36, 37, 37

simulate.mrwd, 41

StMoMo, 3, 6, 19, 21–23, 35, 41

StMoMo-package (StMoMo), 41