

# Package ‘elect’

May 8, 2026

**Type** Package

**Title** Estimation of Life Expectancies Using Multi-State Models

**Version** 1.2

**Author** Ardo van den Hout and Mei Sum Chan

**Maintainer** Ardo van den Hout <Ardo.vandenHout@ucl.ac.uk>

**Description** Functions to compute state-specific and marginal life expectancies. The computation is based on a fitted continuous-time multi-state model that includes an absorbing death state; see Van den Hout (2017, ISBN:9781466568402). The fitted multi-state model should be estimated using the 'msm' package using age as the time-scale.

**Depends** msm, nnet

**License** GPL-2

**Encoding** UTF-8

**LazyData** true

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2019-04-03 17:40:03 UTC

## Contents

check.RestrAndConst . . . . .	2
elect . . . . .	3
electData . . . . .	5
explore . . . . .	5
hazards . . . . .	6
plot.elect . . . . .	8
plusmin . . . . .	9
summary.elect . . . . .	10

**Index** [11](#)

---

check.RestrAndConst    *Checking the definition of RestrAndConst in elect*

---

## Description

Function to check the definition of RestrAndConst in the elect call.

## Usage

```
check.RestrAndConst(x, RestrAndConst, PRINT = FALSE)
```

## Arguments

x	Fitted msm model
RestrAndConst	Vector which indexes the independent parameters in model\sopt\spar w.r.t. to the model parameters
PRINT	TRUE for printing the comparison

## Details

Function to help defining the parameter constraints in the elect call when the constraint option is used in msm.

## Value

TRUE when RestrAndConst is well-defined, FALSE otherwise.

## Author(s)

Ardo van den Hout

## See Also

[elect](#)

## Examples

```
# Fit model with constraints:
Q      <- rbind(c(0,0.01,0.01), c(0,0,0.01), c(0,0,0))
model <- msm(state~age, subject = id, data = electData,
             center = FALSE, qmatrix = Q, deathexact = TRUE,
             fixedpars = c(7,8), covariates = ~age+x)
RestrAndConst <- c(1:6,0,0,7)
check.RestrAndConst(model, RestrAndConst, PRINT= TRUE)

# Usage: elect(x = model,..., RestrAndConst = RestrAndConst)
```

---

elect *State-specific and marginal life expectancies*

---

### Description

Estimation of state-specific and marginal life expectancies given a multi-state survival model fitted using the `msm`-package

### Usage

```
elect(x, b.covariates, statedistdata, time.scale.msm = "years",
      h, age.max, S = 0, setseed = NULL, RestrAndConst = NULL,
      statedist.covariates = "age", method = "step")
```

### Arguments

<code>x</code>	Fitted <code>msm</code> model
<code>b.covariates</code>	List with specified covariates values (ignore intercept)
<code>statedistdata</code>	Data used to estimate prevalence distribution of living states
<code>time.scale.msm</code>	Time scale in multi-state model: "years", "months", "weeks", or a value in (0,1]
<code>h</code>	Grid parameter for integration where scale is <code>time.scale.msm</code>
<code>age.max</code>	Assumed maximum age in same time scale as in fitted model
<code>S</code>	Number of replications for estimation of uncertainty ( $S=0$ for no estimation)
<code>setseed</code>	Seed for the random number generation in the simulation
<code>RestrAndConst</code>	Vector which indexes the independent model parameters. Only needed when constraint is used in <code>msm</code> call
<code>statedist.covariates</code>	Names of covariates for model for prevalence distribution of living states
<code>method</code>	Approximation of integral: "step" for simple step function, "MiddleRiemann" or "Simpson"

### Details

The `elect`-package is an add-on to the `msm`-package for models with one death state. In the `msm` call for fitting the model use `center=FALSE`, and names `state` and `age`. Do not use variables encoded as factor by R. Covariate `age` should be the first entry in `b.covariates`. The other covariates in `b.covariates` should follow the order in the `msm` call. The life expectancies are computed by approximating the integral numerically with a grid defined by `h`. The specification of `statedist.covariates` should be a subset of `b.covariates`.

**Value**

A list containing the following components:

pnt	Life expectancies derived from MLE of model parameters
sim	Simulated life expectancies using the MLE of model parameters
h	As specified in <code>elect</code> call
covars	Covariates as specified in <code>elect</code> call
S	S as specified in <code>elect</code> call
sd.model	Fitted model for the prevalence distribution of living states

**Author(s)**

Ardo van den Hout and Mei Sum Chan

**References**

Jackson, C.H. (2011). Multi-State Models for Panel Data: The `msm` Package for R., *Journal of Statistical Software*, 38(8), 1-29.

Van den Hout, A. (2017). *Multi-State Survival Models for Interval-Censored Data*. Boca Raton: CRC/Chapman & Hall.

**See Also**

[summary.elect](#), [plot.elect](#)

**Examples**

```
# Fit msm model:
Q      <- rbind(c(0,0.01,0.01), c(0,0,0.01), c(0,0,0))
model <- msm(state~age, subject = id, data = electData,
             center = FALSE, qmatrix = Q, deathexact = TRUE,
             covariates = ~age+x)

# Estimate life expectancies:
sddata <- electData[electData$bsline == 1,]
LEs    <- elect(x = model, b.covariates = list(age = 0, x = 0),
              statedistdata = sddata, h = 0.5, age.max = 50, S = 25)
summary(LEs)
plot(LEs)      # For smoother graphs, increase S
```

---

electData	<i>Three-state data for exploring elect</i>
-----------	---

---

**Description**

Simulated longitudinal data for a three-state illness-death process.

**Usage**

```
electData
```

**Format**

Data frame with 764 rows, grouped by 150 individuals. Simulated interval-censored transition times for living states 1 and 2, and exact times for death state 3. Variables: `id` = identification number, `state` = state, `age` = age in years on a shifted scale (current age minus 70), `x` = binary time-independent covariate (can be interpreted as 0/1 for women/men), `bsline` = baseline record indicator.

**Author(s)**

Ardo van den Hout

**See Also**

[elect](#)

**Examples**

```
# Sample size:
print(length(unique(electData$id)))
# Frequencies number of observation per individual:
print(table(table(electData$id)))
# State table
print(statetable.msm(state, id, data = electData))
```

---

explore	<i>Data statistics for an age-dependent model</i>
---------	---

---

**Description**

Data statistics that are important for fitting an age-dependent multi-state model. Basic summaries and additional information on the age distribution in the data

**Usage**

```
explore(data = NULL, id = NULL, state = NULL, age = NULL,
        digits = 3, HIST = TRUE, hist.col = c("green", "red",
        "blue"), INFO = FALSE)
```

**Arguments**

data	Data frame with variables id, state, and age
id	Identifier. Specify if data is not provided
state	State variable. Specify if data is not provided
age	Age or transformed age. Specify if data is not provided
digits	Number of digits in the output
HIST	TRUE for histograms of the age distribution. FALSE otherwise
hist.col	Colours for the three histograms of the age distribution
INFO	TRUE for returning a list which links id with time intervals between the subsequent records. FALSE otherwise

**Value**

intervals	Data frame with variables interval.length and corresponding id (when INFO = TRUE)
-----------	---

**Author(s)**

Ardo van den Hout. With thanks to Ying Lou.

**Examples**

```
explore(electData)
```

---

hazards	<i>Age-dependent hazards based on a 'msm' model</i>
---------	---

---

**Description**

Graph with transition-specific hazard functions derived from an age-dependent model fitted using 'msm'

**Usage**

```
hazards(x, b.covariates, no.years, trans = NULL,
        max.haz = .5, min.haz = 0, CI = FALSE, col = NULL,
        lty = NULL, lwd = NULL, LEGEND = TRUE,
        location = "topleft", age.shift = 0)
```

**Arguments**

<code>x</code>	Fitted msm model with age as the Gompertz time scale
<code>b.covariates</code>	List with specified covariates values for the prediction (ignore intercept)
<code>no.years</code>	Number of years for the prediction
<code>trans</code>	Matrix with rows (r,s) for hazard of going from state r to state s. Default to all (r,s)-hazards that are modelled in x
<code>max.haz</code>	Upperbound hazard-axis
<code>min.haz</code>	Lowerbound hazard-axis
<code>CI</code>	TRUE for plotting 95% confidence bands. FALSE otherwise
<code>col</code>	Colour for each hazard curve. Example for two curves: <code>col = c("red", "green")</code>
<code>lty</code>	Line type for each hazard curve. Example for two curves: <code>lty = c(1, 2)</code>
<code>lwd</code>	Width of line for each hazard curve. Example for two curves: <code>lwd = c(2, 2)</code>
<code>LEGEND</code>	TRUE for adding a legend. FALSE otherwise
<code>location</code>	Location for legend. Default to "topleft". See help file for legend for further details
<code>age.shift</code>	Value to shift the age scale in the graph. Useful when age in the model is on a shifted scale. Default to 0.

**Details**

This function is an add-on to the functionality in the 'msm' package. A Gompertz model with age as the time scale can be fitted in 'msm' piecewise-constantly by adding age as a covariate. The function `qmatrix.msm` is used by `hazards` repeatedly to take into account the age dependence when calculating the hazards.

**Author(s)**

Ardo van den Hout. With thanks to Ying Lou.

**References**

Jackson, C.H. (2011). Multi-State Models for Panel Data: The msm Package for R., *Journal of Statistical Software*, 38(8), 1-29.

**Examples**

```
# Fit msm model:
Q <- rbind(c(0,0.01,0.01), c(0,0,0.01), c(0,0,0))
model <- msm(state~age, subject = id, data = electData,
             center = FALSE, qmatrix = Q, deathexact = TRUE,
             covariates = ~age+x)
# Plot hazards on original age scale:
hazards(model, b.covariates = list(age = 0, x = 0),
         no.years = 20, max.haz = 0.25, age.shift = -70)
```

---

`plot.elect`*Plotting of life expectancies*

---

**Description**

Graphical representation by smoothed densities of the life expectancies as estimated by `elect`

**Usage**

```
## S3 method for class 'elect'  
plot(x, which = NULL, kernel = "gaussian", col = "red", lwd = 2, cex.lab = 1, ...)
```

**Arguments**

<code>x</code>	Life expectancies estimated by <code>elect</code>
<code>which</code>	Subselection for plotting (following the order in <code>summary</code> ). Example: <code>which = c(1,3,5)</code>
<code>kernel</code>	Character string for smoothing kernel ("gaussian", "rectangular", "triangular", "epanechnikov", "biweight", "cosine", or "optcosine")
<code>col</code>	Colour of curves
<code>lwd</code>	Line width of curves
<code>cex.lab</code>	Magnification to be used for axis-labels
<code>...</code>	Other arguments (not yet implemented)

**Details**

Presents distributions of the estimated life expectancies derived from the maximum likelihood estimate of the model parameters. The smoothing is undertaken using the R function `density`.

**Author(s)**

Ardo van den Hout

**See Also**

[elect](#)

---

plusmin

*Computation of a function of life expectancies*


---

**Description**

Compute an additive function of a series of estimated life expectancies

**Usage**

```
plusmin(x, index = NA, func = "plus", probs = c(0.025, 0.5, 0.975), digits = 3)
```

**Arguments**

x	Life expectancies estimated by elect
index	Selection of the life expectancies in the function (following the order in summary)
func	Required series of "plus" and "minus". Example: func = c("plus", "plus")
probs	Probabilities for the quantiles
digits	Number of digits in the output

**Details**

This function requires that uncertainty of LEs is estimated; that is, S is not 0.

**Author(s)**

Ardo van den Hout

**See Also**

[elect](#)

**Examples**

```
# Fit msm model:
Q    <- rbind(c(0,0.01,0.01), c(0,0,0.01), c(0,0,0))
model <- msm(state~age, subject = id, data = electData,
             center = FALSE, qmatrix = Q, deathexact = TRUE,
             covariates = ~age+x)

# Estimate function of life expectancies:
sddata <- electData[electData$bsline == 1,]
LEs    <- elect(x = model, b.covariates = list(age = 0, x = 1),
              statedistdata = sddata, h = 0.5, age.max = 50, S = 25)
plusmin(LEs, index = c(2,4), func = "plus")
```

---

summary.elect	<i>Summarises the estimation of the life expectancies</i>
---------------	---

---

**Description**

Summary of estimated state-specific and marginal life expectancies

**Usage**

```
## S3 method for class 'elect'  
summary(object, probs = c(0.025, 0.5, 0.975),  
         digits = 3, StartStateTotals = FALSE,  
         print = TRUE, sd.model = FALSE,...)
```

**Arguments**

object	Life expectancies estimated by elect
probs	Numeric vector of probabilities with values in [0,1] for quantiles
digits	Number of decimal places in output
StartStateTotals	TRUE for output on start-state totals e_r. (for S>0)
print	TRUE for printing output to screen, FALSE otherwise
sd.model	TRUE for printing a summary of the fitted model for the prevalence
...	Other arguments (not yet implemented)

**Details**

A summary for the state-specific and marginal life expectancies as derived in elect. Quantiles are derived from simulation based on the maximum likelihood estimation.

**Author(s)**

Ardo van den Hout

**See Also**

[elect](#), [plusmin](#), [plot.elect](#)

# Index

`check.RestrAndConst`, 2

`elect`, 2, 3, 5, 8–10

`electData`, 5

`explore`, 5

`hazards`, 6

`plot (plot.elect)`, 8

`plot.elect`, 4, 8, 10

`plusmin`, 9, 10

`summary (summary.elect)`, 10

`summary.elect`, 4, 10