

# Package ‘MortCast’

October 12, 2022

**Type** Package

**Title** Estimation and Projection of Age-Specific Mortality Rates

**Version** 2.7-0

**Date** 2022-03-31

**Author** Hana Sevcikova, Nan Li and Patrick Gerland

**Maintainer** Hana Sevcikova <hanas@uw.edu>

**Description** Age-specific mortality rates are estimated and projected using the Kannisto, Lee-Carter and related methods as described in Sevcikova et al. (2016) <[doi:10.1007/978-3-319-26603-9\\_15](https://doi.org/10.1007/978-3-319-26603-9_15)>.

**License** GPL (>= 2)

**Depends** R (>= 3.5.0), wpp2017

**RoxygenNote** 7.1.1

**LazyLoad** True

**LazyData** True

**NeedsCompilation** yes

**Repository** CRAN

**Date/Publication** 2022-03-31 21:30:02 UTC

## R topics documented:

MortCast-package . . . . .	2
cokannisto . . . . .	4
cokannisto.estimate . . . . .	5
kannisto . . . . .	7
kannisto.estimate . . . . .	8
kannisto.predict . . . . .	9
leecarter.estimate . . . . .	10
life.table . . . . .	12
lileecarter.estimate . . . . .	14
logquad . . . . .	15
LQcoef . . . . .	17

mlt . . . . .	18
MLTlookup . . . . .	19
mortcast . . . . .	20
mortcast.blend . . . . .	22
pmd . . . . .	25
PMDadjcoef . . . . .	28
PMDrho . . . . .	29
rotate.leecarter . . . . .	30
<b>Index</b>	<b>32</b>

---

MortCast-package	<i>MortCast: Estimation and Projection of Age-Specific Mortality Rates</i>
------------------	----------------------------------------------------------------------------

---

## Description

Age-specific mortality rates are estimated and projected using the Kannisto, Lee-Carter and related methods as described in Sevcikova et al. (2016) <doi:10.1007/978-3-319-26603-9\_15>.

## Details

The package implements methodology described in Sevcikova et al. (2016) that is related to estimating and predicting age-specific mortality rates. The main functions are:

- `cokannisto`: Extrapolates given mortality rates into higher ages using the Coherent Kannisto method. The original Kannisto method (with sex-independent extrapolation) is available in the function `kannisto`.
- `lileecarter.estimate`: Estimates the coherent Lee-Carter parameters for male and female mortality rates (Li and Lee 2005), i.e. sex-independent parameters  $a_x$  and  $k_t$ , and the coherent parameter  $b_x$ . In addition, it computes the ultimate  $b_x^u$  for rotation (Li et al. 2013). The underlying sex-independent estimation is implemented in the function `leecarter.estimate`.
- `mortcast`: Using estimated coherent Lee-Carter parameters and given future sex-specific life expectancies, it projects age-specific mortality rates, while (by default) rotating the  $b_x$  parameter as described in Li et al. (2013).

Functions contained in the package can be used to apply Algorithm 2 in Sevcikova et al. (2016) as shown in the Example below. It can be used for both, 5-year and 1-year age groups.

Other methods for forecasting mortality rates are available:

- `pmd`: pattern of mortality decline
- `mlt`: model life tables
- `logquad`: log-quadratic mortality model
- `mortcast.blend`: combining two different methods

A life table can be constructed using the `life.table` function.

**Author(s)**

Hana Sevcikova, Nan Li and Patrick Gerland

**References**

Li, N. and Lee, R. D. (2005). Coherent mortality forecasts for a group of populations: An extension of the Lee-Carter method. *Demography*, 42, 575-594.

Li, N., Lee, R. D. and Gerland, P. (2013). Extending the Lee-Carter method to model the rotation of age patterns of mortality decline for long-term projections. *Demography*, 50, 2037-2051.

Sevcikova H., Li N., Kantorova V., Gerland P., Raftery A.E. (2016). Age-Specific Mortality and Fertility Rates for Probabilistic Population Projections. In: Schoen R. (eds) *Dynamic Demographic Analysis*. The Springer Series on Demographic Methods and Population Analysis, vol 39. Springer, Cham. [Earlier version](#).

**Examples**

```
# This example applies Algorithm 2 in Sevcikova et al. (2016)
# on data from WPP2017 for China
#
data(mxM, mxF, e0Fproj, e0Mproj, package = "wpp2017")
country <- "China"

# extract observed mortality rates for male and female
mxm <- subset(mxM, name == country)[,4:16]
mxf <- subset(mxF, name == country)[,4:16]
rownames(mxm) <- rownames(mxf) <- c(0,1, seq(5, 100, by=5))

# Step 1: extrapolate from 100+ to 130+ using Coherent Kannisto
mx130 <- cokannisto(mxm, mxf)

# Steps 2-5: estimate coherent Lee-Carter parameters
# (here ax is computed from the last observed period
# and smoothened over ages)
lc.est <- lileecarter.estimate(mx130$male, mx130$female,
                             ax.index = ncol(mx130$male), ax.smooth = TRUE)

# Steps 6-9: project future mortality rates based on future
# life expectancies from WPP2017
e0f <- as.numeric(subset(e0Fproj, name == country)[-(1:2)])
e0m <- as.numeric(subset(e0Mproj, name == country)[-(1:2)])
names(e0f) <- names(e0m) <- colnames(e0Fproj)[-(1:2)]
pred <- mortcast(e0m, e0f, lc.est)

# plot projection for the first and last future time period
plot(pred$female$mx[, "2015-2020"], type="l", log="y",
      ylim=range(pred$female$mx, pred$male$mx), xaxt="n",
      ylab="mx", xlab="Age", main=country, col="red")
axis(1, at=1:nrow(pred$female$mx),
      labels=rownames(pred$female$mx))
lines(pred$male$mx[, "2015-2020"], col="blue")
lines(pred$female$mx[, "2095-2100"], col="red", lty=2)
```

```
lines(pred$male$mx[, "2095-2100"], col="blue", lty=2)
legend("topleft", legend=c("male 2015-2020", "female 2015-2020",
  "male 2095-2100", "female 2095-2100"), bty="n",
  col=rep(c("blue", "red"),2), lty=c(1,1,2,2))
```

---

cokannisto

*Coherent Kannisto Method*


---

## Description

Extrapolate given mortality rates into higher ages using the Coherent Kannisto method as described in Sevcikova et al. (2016).

## Usage

```
cokannisto(
  mxM,
  mxF,
  est.ages = seq(80, 95, by = 5),
  proj.ages = seq(100, 130, by = 5)
)
```

## Arguments

mxM	A vector or matrix of male mortality rates. If it is a matrix, rows correspond to age groups with rownames identifying the corresponding age as integers. By default five-years age groups are assigned to rows if rownames are not given.
mxF	A vector or matrix of female mortality rates. Its length or dimension should be the same mxM.
est.ages	A vector of integers identifying the ages to be used for estimation. It should be a subset of rownames of mxM. Change the defaults if 1-year age groups are used (see Example in <a href="#">kannisto</a> ).
proj.ages	A vector of integers identifying the age groups for which mortality rates are to be projected. Change the defaults if 1-year age groups are used (see Example in <a href="#">kannisto</a> ).

## Details

The function first estimates the coherent Kannisto parameters by passing mortality rates for age groups `est.ages` into the `cokannisto.estimate` function. The estimated parameters are then passed to the projection function `kannisto.predict` to extrapolate into ages `proj.ages`. Lastly, the input mortality objects are extended by results for the extrapolated ages. If `proj.ages` contains age groups that are included in `mxM` and `mxF`, values for those age groups are overwritten.

**Value**

A list of two vectors or matrices (for male and female) containing the input mortality objects extended by the extrapolated age groups.

**References**

Sevcikova H., Li N., Kantorova V., Gerland P., Raftery A.E. (2016). Age-Specific Mortality and Fertility Rates for Probabilistic Population Projections. In: Schoen R. (eds) Dynamic Demographic Analysis. The Springer Series on Demographic Methods and Population Analysis, vol 39. Springer, Cham

**See Also**

[cokannisto.estimate](#), [kannisto.predict](#)

**Examples**

```
data(mxM, mxF, package = "wpp2017")
country <- "South Africa"
mxm <- subset(mxM, name == country)[,-(1:3)]
mxf <- subset(mxF, name == country)[,-(1:3)]
rownames(mxm) <- rownames(mxf) <- c(0,1, seq(5, 100, by=5))
mxnew <- cokannisto(mxm, mxf)
ages <- as.integer(rownames(mxnew$male))
plot(ages, mxnew$male[, "2095-2100"], type="l", log="y",
     xlab="age", ylab="mx", col="blue", main=country)
lines(ages, mxnew$female[, "2095-2100"], col="red")
lines(ages, mxnew$male[, "2010-2015"], lty=2, col="blue")
lines(ages, mxnew$female[, "2010-2015"], lty=2, col="red")
legend("bottomright", legend=c("male 2010-2015", "female 2010-2015",
                              "male 2095-2100", "female 2095-2100"), bty="n",
      col=rep(c("blue", "red"),2), lty=c(2,2,1,1))
```

---

cokannisto.estimate     *Coherent Kannisto Estimation*

---

**Description**

Estimate the coherent Kannisto parameters as described in Sevcikova et al. (2016).

**Usage**

```
cokannisto.estimate(mxM, mxF, ages, fitted = TRUE)
```

**Arguments**

mxM	A vector of male mortality rates.
mxF	A vector of female mortality rates.
ages	A vector of ages corresponding to mxM and mxF.
fitted	Logical. If TRUE the fitted values and residuals are returned.

**Details**

Given the Kannisto equation  $\text{logit}(m_x) = \log(c) + dx$ , the Coherent Kannisto method estimates the  $d$  parameter jointly for male and female data, in order to prevent mortality crossovers in higher ages.

**Value**

List of two lists, one for male and one for female. Each of the two lists contains the following components:

**coefficients:** named vector with the coherent Kannisto coefficients  $c$  and  $d$ . The  $d$  values are the same in both lists.

**fitted.values:** the fitted values (not included if `fitted` is FALSE)

**residuals:** input rates minus the fitted values (not included if `fitted` is FALSE)

**References**

Sevcikova H., Li N., Kantorova V., Gerland P., Raftery A.E. (2016). Age-Specific Mortality and Fertility Rates for Probabilistic Population Projections. In: Schoen R. (eds) Dynamic Demographic Analysis. The Springer Series on Demographic Methods and Population Analysis, vol 39. Springer, Cham

**See Also**

[cokannisto](#), [kannisto.predict](#), [kannisto](#)

**Examples**

```
data(mxM, mxF, package = "wpp2017")
country <- "Brazil"
mxm <- subset(mxM, name == country)[,"2010-2015"]
mxf <- subset(mxF, name == country)[,"2010-2015"]
cokannisto.estimate(mxm[18:21], mxf[18:21], ages = 18:21)
```

---

`kannisto`*Kannisto Method*

---

## Description

Extrapolate given mortality rates using the original Kannisto method.

## Usage

```
kannisto(mx, est.ages = seq(80, 95, by = 5), proj.ages = seq(100, 130, by = 5))
```

## Arguments

<code>mx</code>	A vector or matrix of mortality rates. If it is a matrix, rows correspond to age groups with rownames identifying the corresponding age as integers. By default five-years age groups are assigned to rows if rownames are not given.
<code>est.ages</code>	A vector of integers identifying the ages to be used for estimation. It should be a subset of rownames of <code>mx</code> . Change the defaults if 1-year age groups are used (see Example).
<code>proj.ages</code>	A vector of integers identifying the age groups for which mortality rates are to be projected. Change the defaults if 1-year age groups are used (see Example).

## Details

The function first estimates the original Kannisto parameters by passing mortality rates for age groups `est.ages` into the `kannisto.estimate` function. The estimated parameters are then passed to the projection function `kannisto.predict` to extrapolate into ages `proj.ages`. Lastly, the input mortality object is extended by results for the extrapolated ages. If `proj.ages` contains age groups that are included in `mx`, values for those age groups are overwritten.

## Value

A vector or matrix containing the input mortality object `mx` extended by the extrapolated age groups.

## References

Thatcher, A. R., Kannisto, V. and Vaupel, J. W. (1998). The Force of Mortality at Ages 80 to 120, volume 5 of Odense Monographs on Population Aging Series. Odense, Denmark: Odense University Press.

## See Also

[kannisto.estimate](#), [kannisto.predict](#), [cokannisto](#)

## Examples

```

data(mxM, package = "wpp2017")
mx <- subset(mxM, name == "Burkina Faso"),-(1:3)]
rownames(mx) <- c(0,1, seq(5, 100, by=5))
mxnew <- kannisto(mx)
ages <- as.integer(rownames(mxnew))
plot(ages, mxnew[, "2095-2100"], type="l", log="y",
     xlab="age", ylab="mx", col="red")
lines(ages, mxnew[, "2010-2015"])

# Kannisto for 1-year age groups
# derive toy 1-year mx using model life tables at e0 of 70
mx1y <- mlt(70, sex = "male", nx = 1)
# Pretend we only observed mx for ages 0:100.
# Use 90-99 for estimation and extend mx from 100 to 140
mx1ynew <- kannisto(mx1y[1:100, , drop = FALSE], est.ages = 90:99, proj.ages = 100:140)
# Plot the new mx for old ages
plot(80:140, mx1ynew[81:141], type = "l", xlab="age", ylab="mx", col="red")
# Check how it compares to the original mx that was not used in the estimation
lines(100:130, mx1y[101:nrow(mx1y)])

```

---

kannisto.estimate

*Kannisto Estimation*


---

## Description

Estimate the Kannisto parameters (Thatcher et al. 1998).

## Usage

```
kannisto.estimate(mx, ages)
```

## Arguments

mx	A vector of mortality rates.
ages	A vector of ages corresponding to mx. These can be indices of age groups or raw ages.

## Details

Given the Kannisto equation  $\text{logit}(m_x) = \log(c) + dx$ , the function estimates the  $c$  and  $d$  parameters using values of ages as the covariate  $x$ .



**Value**

List with the following components:

`coefficients`: named vector with Kannisto coefficients  $c$  and  $d$ .

`fitted.values`: the fitted values

`residuals`: input rates minus the fitted values

**References**

Thatcher, A. R., Kannisto, V. and Vaupel, J. W. (1998). The Force of Mortality at Ages 80 to 120, volume 5 of Odense Monographs on Population Aging Series. Odense, Denmark: Odense University Press.

**See Also**

[kannisto.predict](#), [kannisto](#), [cokannisto.estimate](#)

**Examples**

```
data(mxM, package = "wpp2017")
mx <- subset(mxM, name == "Canada")[, "2010-2015"]
kannisto.estimate(mx[18:21], ages = 18:21)
```

---

kannisto.predict	<i>Kannisto Prediction</i>
------------------	----------------------------

---

**Description**

Given estimated Kannisto parameters (coherent or original), it predicts mortality rates for given ages.

**Usage**

```
kannisto.predict(pars, ages)
```

**Arguments**

pars	A named vector with Kannisto coefficients $c$ and $d$ (e.g. result of <a href="#">kannisto.estimate</a> or <a href="#">cokannisto.estimate</a> ).
ages	A vector of ages to make prediction for. These can be indices of age groups or raw ages, but on the same scale as used in the estimation.

**Details**

Given parameters  $c$  and  $d$  in `pars`, the function uses the Kannisto equation  $\text{logit}(m_x) = \log(c) + dx$ , to predict mortality rates for age groups  $x$  given by ages.

**Value**

Vector of predicted mortality rates.

**References**

Thatcher, A. R., Kannisto, V. and Vaupel, J. W. (1998). The Force of Mortality at Ages 80 to 120, volume 5 of Odense Monographs on Population Aging Series. Odense, Denmark: Odense University Press.

**See Also**

[cokannisto](#), [kannisto.estimate](#), [cokannisto.estimate](#)

**Examples**

```
data(mxM, mxF, package = "wpp2017")
mxm <- subset(mxM, name == "Germany")[, "2010-2015"]
ages <- c(0, 1, seq(5, 130, by=5))

# using original Kannisto parameters
pars <- kannisto.estimate(mxm[18:21], ages = ages[18:21])
mxm.pred <- kannisto.predict(pars$coefficients, ages = ages[22:28])
plot(ages, c(mxm[1:21], mxm.pred), type="l", log="y",
      xlab="age", ylab="mx")

# Coherent Kannisto
mxf <- subset(mxF, name == "Germany")[, "2010-2015"]
copars <- cokannisto.estimate(
  mxm[18:21], mxf[18:21], ages = ages[18:21])
cmxm.pred <- kannisto.predict(copars[["male"]]$coefficients, ages = ages[22:28])
cmxf.pred <- kannisto.predict(copars[["female"]]$coefficients, ages = ages[22:28])
plot(ages, c(mxm[1:21], cmxm.pred), type="l", log="y",
      xlab="age", ylab="mx", col="blue")
lines(ages, c(mxf[1:21], cmxf.pred), col="red")
```

---

leecarter.estimate      *Lee-Carter Estimation*

---

**Description**

Estimate Lee-Carter parameters (Lee and Carter 1992).

**Usage**

```
leecarter.estimate(
  mx,
  ax.index = NULL,
  ax.smooth = FALSE,
```

```

    ax.smooth.df = NULL,
    bx.postprocess = TRUE,
    nx = 5
  )

```

### Arguments

mx	A matrix of age-specific mortality rates where rows correspond to age groups and columns correspond to time periods. Rownames define the starting ages of the age groups.
ax.index	A vector of column indices of mx to be used to estimate the $a_x$ parameter. By default all time periods are used.
ax.smooth	Logical allowing to smooth the $a_x$ over ages.
ax.smooth.df	Degree of freedom for smoothing if ax.smooth is TRUE. Default is half the length of $a_x$ .
bx.postprocess	Logical determining if numerical anomalies in $b_x$ should be dealt with.
nx	Size of age groups. By default ages are determined by rownames of mx. This argument is only used if mx has no rownames. If nx is 5, the age groups are interpreted as 0, 1, 5, 10, .... For nx equals 1, the age groups are interpreted as 0, 1, 2, 3, ....

### Details

The function estimates parameters of  $\log(m_x(t)) = a_x + b_x k(t) + \epsilon_x(t)$  (Lee and Carter 1992). The argument ax.index determines which time periods to use to estimate the  $a_x$  parameter, while ax.smooth controls if the resulting  $a_x$  should be smoothed over ages (see Sevcikova et al. 2016 for details).

### Value

List with elements ax, bx and kt corresponding to the estimated parameters.

### References

Lee, R. D. and Carter, L. (1992). Modeling and forecasting the time series of US mortality. Journal of the American Statistical Association, 87, 659-671.

Sevcikova H., Li N., Kantorova V., Gerland P., Raftery A.E. (2016). Age-Specific Mortality and Fertility Rates for Probabilistic Population Projections. In: Schoen R. (eds) Dynamic Demographic Analysis. The Springer Series on Demographic Methods and Population Analysis, vol 39. Springer, Cham

### See Also

[mortcast](#), [lileecarter.estimate](#)

**Examples**

```

data(mxM, package = "wpp2017")
mx <- subset(mxM, name == "Netherlands")[,4:16]
rownames(mx) <- c(0,1, seq(5, 100, by=5))
lc.ax.avg <- leecarter.estimate(mx)
lc.ax.last <- leecarter.estimate(mx, ax.index=ncol(mx))
plot(lc.ax.avg$ax, type="l")
lines(lc.ax.last$ax, col="blue")

```

life.table

*Life Table Function***Description**

Function for obtaining life table quantities from mortality rates.

**Usage**

```

life.table(
  mx,
  sex = c("male", "female", "total"),
  abridged = TRUE,
  a0rule = c("ak", "cd"),
  radix = 1,
  open.age = 130
)

```

**Arguments**

mx	Vector of age-specific mortality rates $nm_x$ . If abridged is TRUE (default), the elements correspond to $1m_0, 4m_1, 5m_5, 5m_{10}, \dots$ . If abridged is FALSE, they correspond to $1m_0, 1m_1, 1m_2, 1m_3, \dots$ .
sex	Which sex the mortality rates correspond to.
abridged	Is it an abridged life table (TRUE, default) or not (FALSE). In the former case, the $mx$ vector is interpreted as corresponding to age groups 0, 1-4, 5-9, 10-14, $\dots$ . If FALSE, the $mx$ vector is interpreted as corresponding to one-year age groups, i.e. 0, 1, 2, 3, $\dots$ .
a0rule	Rule for approximation of $a_0$ . "ak" (default) uses the Andreev-Kingkade method (Andreev and Kingkade, 2015), "cd" uses the Coale-Demeany method.
radix	Base of the life table.
open.age	Open age group. If smaller than the last age group of $mx$ , the life table is truncated. It does not have any effect if larger than the last age group.

## Details

Computes a life table corresponding to given mortality rates for either 5- or 1-year age groups. The implementation follows Preston et al. (2001).

## Value

Data frame with rows corresponding to age groups and the following columns:

**age** Starting year of the age group.

**mx** Age-specific mortality rates as passed into the `mx` argument.

**qx** Probability of dying between ages  $x$  and  $x+n$ .

**lx** Number of survivors at age  $x$ .

**dx** Number of deaths between ages  $x$  and  $x+n$ .

**Lx** Person-years lived between ages  $x$  and  $x+n$ .

**sx** Survival rate from age  $x$  to  $x+n$ . Note that in an abridged life table, `sx` always refers to 5-year intervals. Here, `sx` in the first row is the survival from births to the second age group, `sx` in the second row is the survival from age 0-4 to age 5-9, third row has the survival from 5-9 to 10-14 etc.

**Tx** Person-years lived after age  $x$ .

**ex** Life expectancy at age  $x$ .

**ax** Average person-years lived in the interval by those dying in the interval. For young ages, it follows Preston et al. (2001), Table 3.3 on page 48. If `a0rule` is "ak" (default) the Andreev-Kingkade method is used for `a0`. For compatibility with computations done at the UN, we set `ax` for ages 5 and 10 in the abridged version to 2.5. For an unabridged life table, `ax` is set to 0.5 for all but first and last age groups.

## References

Preston, S.H., Heuveline, P., Guillot, M. (2001). *Demography: Measuring and Modeling Population Processes*. Oxford: Blackwell Publishers Ltd.

Andreev, E.M. and Kingkade, W.W. (2015). [Average age at death in infancy and infant mortality level: Reconsidering the Coale-Demeny formulas at current levels of low mortality](#). *Demographic Research*, 33(13), p.363-390. DOI: 10.4054/DemRes.2015.33.13

## Examples

```
data(mxF, e0Fproj, package = "wpp2017")
# get female mortality of Mexico for the current year
country <- "Mexico"
mxf <- subset(mxF, name == country)[,"2010-2015"]
life.table(mxf, sex = "female")
```

---

lileecarter.estimate *Coherent Lee-Carter Estimation*

---

## Description

Estimate coherent Lee-Carter parameters (Li and Lee 2005).

## Usage

```
lileecarter.estimate(mxM, mxF, nx = 5, ...)
```

## Arguments

mxM	A matrix of male age-specific mortality rates where rows correspond to age groups and columns correspond to time periods. For 5-year age groups, the first and second rows corresponds to 0-1 and 1-5 age groups, respectively. Row-names define the starting ages of the respective groups.
mxF	A matrix of female mortality rates of the same shape as mxM.
nx	Size of age groups. Should be either 5 or 1.
...	Additional arguments passed to <a href="#">leecarter.estimate</a> .

## Details

The coherent Lee-Carter parameters for male and female mortality rates share the same  $b_x$  which is the average of the age-specific  $b_x$  parameters.

The function in addition computes the ultimate  $b_x^u$  as defined in Li et al. (2013) based on the coherent  $b_x$ .

## Value

List containing elements `bx` (coherent  $b_x$  parameter), `ultimate.bx` (ultimate  $b_x^u$  parameter), `ages` (age groups), `nx` (age group interval), and lists `female` and `male`, each with the Lee-Carter parameters.

## References

Li, N. and Lee, R. D. (2005). Coherent mortality forecasts for a group of populations: An extension of the Lee-Carter method. *Demography*, 42, 575-594.

Li, N., Lee, R. D. and Gerland, P. (2013). Extending the Lee-Carter method to model the rotation of age patterns of mortality decline for long-term projections. *Demography*, 50, 2037-2051.

**Examples**

```

data(mxM, mxF, package = "wpp2017")
country <- "Germany"
mxm <- subset(mxM, name == country)[,4:16]
mxf <- subset(mxF, name == country)[,4:16]
rownames(mxm) <- rownames(mxf) <- c(0,1, seq(5, 100, by=5))
lc <- lileecarter.estimate(mxm, mxf)
plot(lc$bx, type="l")
lines(lc$ultimate.bx, lty=2)

```

logquad

*Log-Quadratic Mortality Model***Description**

Predict age-specific mortality rates using the Log-Quadratic Mortality Model (Wilmoth et al. 2012).

**Usage**

```

logquad(
  e0,
  sex = c("male", "female", "total"),
  my.coefs = NULL,
  q5ranges = c(1e-04, 0.9),
  k = 0,
  keep.lt = FALSE,
  ...
)

logquadj(e0m, e0f, ...)

```

**Arguments**

e0	Vector of target life expectancies.
sex	Which sex does the give e0 corresponds to.
my.coefs	Data frame with columns "sex", "age", "ax", "bx", "cx", "vx". The "sex" column should contain values "female", "male" and/or "total". The "age" column must be sorted so that it assures that rows correspond to ages in increasing order. Any NAs are internally converted to zeros. If not given, the dataset <a href="#">LQcoef</a> is used.
q5ranges	A vector of size two, giving the min and max of 5q0 used in the bisection method.
k	Value of the $k$ parameter.
keep.lt	Logical. If TRUE additional life table columns are kept in the resulting object.
...	Additional arguments passed to the underlying function.





---

LQcoef	<i>Coefficients for the Log-Quadratic Mortality Model</i>
--------	-----------------------------------------------------------

---

**Description**

Data object containing a table of coefficients to be used in the Log-Quadratic Model as implemented in the `logquad` function.

**Usage**

```
data(LQcoef)
```

**Format**

Data frame containing columns “sex”, “age”, “ax”, “bx”, “cx”, “vx”. Rows correspond to sex and age groups.

**Source**

<https://u.demog.berkeley.edu/~jrw/LogQuad/>

**References**

Wilmoth, J., Zureick, S., Canudas-Romo, V., Inoue, M., Sawyer, C. (2012). A Flexible Two-Dimensional Mortality Model for Use in Indirect Estimation. *Population studies*, 66(1), 1-28. doi: [10.1080/00324728.2011.611411](https://doi.org/10.1080/00324728.2011.611411)

**See Also**

[logquad](#)

**Examples**

```
data(LQcoef)
head(LQcoef)
```

mlt

*Model Life Tables Mortality Patterns***Description**

Predict age-specific mortality rates using Coale-Demeny and UN model life tables.

**Usage**

```
mlt(e0, sex = c("male", "female"), type = "CD_West", nx = 5, ...)
```

```
mltj(e0m, e0f, ..., nx = 5)
```

**Arguments**

<code>e0</code>	A time series of target life expectancy.
<code>sex</code>	Either "male" or "female".
<code>type</code>	Type of the model life table. Available options are "CD_East", "CD_North", "CD_South", "CD_West", "UN_Chilean", "UN_Far_Eastern", "UN_General", "UN_Latin_American", "UN_South_Asian".
<code>nx</code>	Size of age groups. Should be either 5 or 1.
<code>...</code>	Additional arguments passed to the underlying function.
<code>e0m</code>	A time series of target male life expectancy.
<code>e0f</code>	A time series of target female life expectancy.

**Details**

Given a level of life expectancy ( $e_0$ ), sex and a type of model life table, the function extracts the corresponding mortality pattern from [MLTlookup](#) (for abridged LT) or [MLT1Ylookup](#) (for 1-year LT), while interpolating between neighboring  $e_0$  groups. Function `mlt` is for one sex, while `mltj` can be used for both sexes.

**Value**

Function `mlt` returns a matrix with the predicted mortality rates. Columns correspond to the values in the  $e_0$  vector and rows correspond to age groups. Function `mltj` returns a list of such matrices, one for each sex.

**References**

<https://www.un.org/development/desa/pd/data/extended-model-life-tables>

Coale, A., P. Demeny, and B. Vaughn. 1983. Regional model life tables and stable populations. 2nd ed. New York: Academic Press.

**See Also**

[mortcast](#), [mortcast.blend](#), [pmd](#), [MLTlookup](#)

**Examples**

```
data(e0Fproj, package = "wpp2017")
country <- "Uganda"
# get target e0
e0f <- subset(e0Fproj, name == country)[-(1:2)]
# project into future using life table Cole-Demeny North
mx <- mlt(e0f, sex = "female", type = "CD_North")
# plot first projection in black and the remaining ones in grey
plot(mx[,1], type = "l", log = "y", ylim = range(mx),
      ylab = "female mx", xlab = "Age",
      main = paste(country, "5-year age groups"))
for(i in 2:ncol(mx)) lines(mx[,i], col = "grey")

# MLT for 1-year age groups
mx1y <- mlt(e0f, sex = "female", type = "CD_North", nx = 1)
plot(mx1y[,1], type = "l", log = "y", ylim = range(mx1y),
      ylab = "female mx", xlab = "Age",
      main = paste(country, "1-year age groups"))
for(i in 2:ncol(mx1y)) lines(mx1y[,i], col = "grey")
```

---

MLTlookup

*Model Life Tables Lookup*


---

**Description**

Lookup tables containing values for various model life tables, including Coale-Demeny and UN life tables.

**Usage**

```
data(MLTlookup)
data(MLT1Ylookup)
```

**Format**

Data frame with the following columns:

- type** Type of the model life table. Available options are “CD\_East”, “CD\_North”, “CD\_South”, “CD\_West”, “UN\_Chilean”, “UN\_Far\_Eastern”, “UN\_General”, “UN\_Latin\_American”, “UN\_South\_Asian”. For the CD types, see Coale et al. (1983). For the UN types, see the link in References below.
- sex** Code for distinguishing sexes. 1 is for male, 2 is for female.
- age** Starting age of an age group. In MLTlookup these are 0, 1, 5, 10, ... 130. The MLT1Ylookup table contains 1-year ages ranging from 0 to 130.

**e0** Level of life expectancy, starting at 20 and going by steps of 2.5 up to 115.

**mx** Mortality rates.

**lx, Lx, sx** Other life table columns.

### Source

An updated version of these datasets were provided by Sara Hertog, United Nations Population Division, in October 2021 (package version  $\geq$  2.6-0). For previous version of the tables, install MortCast 2.5-0: `'devtools::install_github("PPgp/MortCast@v2.5-0")'`

### References

Coale, A., P. Demeny, and B. Vaughn. 1983. Regional model life tables and stable populations. 2nd ed. New York: Academic Press.

<https://www.un.org/development/desa/pd/data/extended-model-life-tables>

### See Also

[mlt](#)

### Examples

```
data(MLTlookup)
str(MLTlookup)
# CD West life table for male at e0 of 80
subset(MLTlookup, type == "CD_West" & sex == 1 & e0 == 80)
```

---

mortcast

*Coherent Rotated Lee-Carter Prediction*

---

### Description

Predict age-specific mortality rates using the coherent rotated Lee-Carter method.

### Usage

```
mortcast(
  e0m,
  e0f,
  lc.pars,
  rotate = TRUE,
  keep.lt = FALSE,
  constrain.all.ages = FALSE,
  ...
)
```

**Arguments**

<code>e0m</code>	A time series of future male life expectancy.
<code>e0f</code>	A time series of future female life expectancy.
<code>lc.pars</code>	A list of coherent Lee-Carter parameters with elements <code>bx</code> , <code>ultimate.bx</code> , <code>ages</code> , <code>nx</code> , <code>female</code> and <code>male</code> as returned by <code>lileecarter.estimate</code> . The <code>female</code> and <code>male</code> objects are again lists that should contain a vector <code>ax</code> and optionally a matrix <code>axt</code> if the $a_x$ parameter needs to be defined as time dependent. In such a case, rows are age groups and columns are time periods corresponding to the length of the <code>e0f</code> and <code>e0m</code> vectors.
<code>rotate</code>	If TRUE the rotation method of $b_x$ is used as described in Li et al. (2013).
<code>keep.lt</code>	Logical. If TRUE additional life table columns are kept in the resulting object.
<code>constrain.all.ages</code>	By default the method constrains the male mortality to be above female mortality for old ages if the male life expectancy is below the female life expectancy. Setting this argument to TRUE causes this constraint to be applied to all ages.
<code>...</code>	Additional life table arguments.

**Details**

This function implements Steps 6-9 of Algorithm 2 in Sevcikova et al. (2016). It uses the abridged or unabridged life table function to find the level of mortality that corresponds to the given life expectancy. Thus, it can be used for both, mortality for 5- or 1-year age groups.

**Value**

List with elements `female` and `male`, each of which contains a matrix `mx` with the predicted mortality rates. If `keep.lt` is TRUE, it also contains matrices `sr` (survival rates), and life table quantities `Lx` and `lx`.

**References**

- Li, N., Lee, R. D. and Gerland, P. (2013). Extending the Lee-Carter method to model the rotation of age patterns of mortality decline for long-term projections. *Demography*, 50, 2037-2051.
- Sevcikova H., Li N., Kantorova V., Gerland P., Raftery A.E. (2016). Age-Specific Mortality and Fertility Rates for Probabilistic Population Projections. In: Schoen R. (eds) *Dynamic Demographic Analysis*. The Springer Series on Demographic Methods and Population Analysis, vol 39. Springer, Cham

**See Also**

[rotate.leecarter](#), [leecarter.estimate](#), [lileecarter.estimate](#), [mortality.blend](#)

**Examples**

```
# estimate parameters from historical mortality data (5-year age groups)
data(mxM, mxF, e0Fproj, e0Mproj, package = "wpp2017")
country <- "Brazil"
```

```

mxm <- subset(mxM, name == country)[,4:16]
mxf <- subset(mxF, name == country)[,4:16]
rownames(mxm) <- rownames(mxf) <- c(0,1, seq(5, 100, by=5))
lc <- lileecarter.estimate(mxm, mxf)

# project into future for given levels of life expectancy
e0f <- as.numeric(subset(e0Fproj, name == country)[-(1:2)])
e0m <- as.numeric(subset(e0Mproj, name == country)[-(1:2)])
pred <- mortcast(e0m, e0f, lc)

# plot first projection in black and the remaining ones in grey
plot(lc$ages, pred$female$mx[,1], type="b", log="y", ylim=range(pred$female$mx),
     ylab="female mx", xlab="Age", main=paste(country, "(5-year age groups)"), cex=0.5)
for(i in 2:ncol(pred$female$mx)) lines(lc$ages, pred$female$mx[,i], col="grey")

# similarly for 1-year age groups
# derive toy 1-year mx using model life tables at given level of e0
mxm1y <- mlt(seq(65, 71, length = 4), sex = "male", nx = 1)
mxf1y <- mlt(seq(73, 78, length = 4), sex = "female", nx = 1)

# estimate parameters
lc1y <- lileecarter.estimate(mxm1y, mxf1y, nx = 1)

# project into the future
pred1y <- mortcast(e0m, e0f, lc1y)

# plot first projection in black and the remaining ones in grey
plot(lc1y$ages, pred1y$female$mx[,1], type="b", log="y", ylim=range(pred1y$female$mx),
     ylab="female mx", xlab="Age", main="1-year age groups", cex=0.5)
for(i in 2:ncol(pred1y$female$mx)) lines(lc1y$ages, pred1y$female$mx[,i], col="grey")

```

---

mortcast.blend

*Mortality Prediction by Method Blending*


---

## Description

Predict age-specific mortality rates using a blend of two different methods (Coherent Lee-Carter, Coherent Pattern Mortality Decline, Log-Quadratic model, or Model Life Tables). Weights can be applied to fine-tune the blending mix.

## Usage

```

mortcast.blend(
  e0m,
  e0f,
  meth1 = "lc",
  meth2 = "mlt",
  weights = c(1, 0.5),

```

```

    nx = 5,
    apply.kannisto = TRUE,
    min.age.groups = 28,
    match.e0 = TRUE,
    keep.lt = FALSE,
    meth1.args = NULL,
    meth2.args = NULL,
    kannisto.args = NULL,
    ...
  )

```

## Arguments

<code>e0m</code>	A time series of future male life expectancy.
<code>e0f</code>	A time series of future female life expectancy.
<code>meth1</code>	Character string giving the name of the first method to blend. It is one of “lc”, “pmd”, “mlt” or “logquad”, corresponding to Coherent Lee-Carter (function <a href="#">mortcast</a> ), Pattern Mortality Decline (function <a href="#">copmd</a> ), Log-Quadratic model (function <a href="#">logquadj</a> ), and Model Life Tables (function <a href="#">mltj</a> ), respectively. The “logquad” method can only be used with 5-year age groups.
<code>meth2</code>	Character string giving the name of the second method to blend. One of the same choices as <code>meth1</code> .
<code>weights</code>	Numeric vector with values between 0 and 1 giving the weight of <code>meth1</code> . If it is a single value, the same weight is applied for all time periods. If it is a vector of size two, it is assumed these are weights for the first and the last time period. Remaining weights will be interpolated. Note that <code>meth2</code> is weighted by $1 - \text{weights}$ .
<code>nx</code>	Size of age groups. Should be either 5 or 1.
<code>apply.kannisto</code>	Logical. If TRUE and if any of the methods results in less than <code>min.age.groups</code> age categories, the coherent Kannisto method ( <a href="#">cokannisto</a> ) is applied to extend the age groups into old ages.
<code>min.age.groups</code>	Minimum number of age groups. Triggers the application of Kannisto, see above. Change the default value if 1-year age groups are used (see Example).
<code>match.e0</code>	Logical. If TRUE the blended <code>mx</code> is scaled so that it matches the input <code>e0</code> .
<code>keep.lt</code>	Logical. If TRUE additional life table columns are kept in the resulting object. Only used if <code>match.e0</code> is TRUE.
<code>meth1.args</code>	List of arguments passed to the function that corresponds to <code>meth1</code> .
<code>meth2.args</code>	List of arguments passed to the function that corresponds to <code>meth2</code> .
<code>kannisto.args</code>	List of arguments passed to the <a href="#">cokannisto</a> function if Kannisto is applied. If 1-year age groups are used various defaults in the Kannisto function need to be changed (see Example).
<code>...</code>	Additional life table arguments.

## Details

The function allows to combine two different methods using given weights. The weights can change over time - by default they are interpolated from the starting weight to the end weight. As the blended mortality rates do not necessarily match the target life expectancy, scaling is applied to improve the match, controlled by the `match.e0` argument. The projection is done for both sexes, so that coherent methods can be applied.

## Value

List with elements `female` and `male`, each of which contains a matrix `mx` with the predicted mortality rates. If the result has been scaled (`match.e0` is TRUE), the element `mx.rawblend` contains the `mx` before scaling. Also in such a case, if `keep.lt` is TRUE, it also contains matrices `sr` (survival rates), and life table quantities `Lx` and `lx`. In addition, the return object contains elements `meth1res` and `meth2res` which contain the results of the functions corresponding to the two methods. Elements `meth1` and `meth2` contain the names of the methods. A vector `weights` contains the final (possibly interpolated) weights.

## See Also

[mortcast](#), [copmd](#), [mltj](#), [logquad](#), [cokannisto](#)

## Examples

```
data(mxM, mxF, e0Fproj, e0Mproj, package = "wpp2017")
country <- "Brazil"
# estimate parameters from historical mortality data
mxm <- subset(mxM, name == country)[,4:16]
mxf <- subset(mxF, name == country)[,4:16]
rownames(mxm) <- rownames(mxf) <- c(0,1, seq(5, 100, by=5))
lcest <- lileecarter.estimate(mxm, mxf)
# project into future
e0f <- subset(e0Fproj, name == country)[-(1:2)]
e0m <- subset(e0Mproj, name == country)[-(1:2)]

# Blend LC and MLT
pred1 <- mortcast.blend(e0m, e0f, meth1 = "lc", meth2 = "mlt",
  meth1.args = list(lc.pars = lcest),
  meth2.args = list(type = "CD_North"),
  weights = c(1,0.25))

# Blend PMD and MLT
pred2 <- mortcast.blend(e0m, e0f, meth1 = "pmd", meth2 = "mlt",
  meth1.args = list(mxm0 = mxm[, "2010-2015"],
    mxf0 = mxf[, "2010-2015"]))

# plot projection by time
plotmx <- function(pred, iage, main)
  with(pred, {
    # blended projections
    plot(female$mx[iage,], type="l",
      ylim=range(meth1res$female$mx[iage,],
```



```

        meth2res$female$mx[iage,]),
        ylab="female mx", xlab="Time", main=main, col = "red")
lines(meth1res$female$mx[iage,], lty = 2)
lines(meth2res$female$mx[iage,], lty = 3)
legend("topright", legend=c("blend", meth1, meth2),
       lty = 1:3, col = c("red", "black", "black"), bty = "n")
    })
age.group <- 3 # 5-9 years old
par(mfrow=c(1,2))
plotmx(pred1, age.group, "LC-MLT (age 5-9)")
plotmx(pred2, age.group, "PMD-MLT (age 5-9)")

# Blend LC and MLT for 1-year age groups
#####
# First interpolate e0 to get 1-year life expectancies (for first five years)
e0m1y <- approx(as.double(e0m[,1:2]), n = 5)$y
e0f1y <- approx(as.double(e0f[,1:2]), n = 5)$y
# derive toy mx in order to get some LC parameters
mxm1y <- mlt(seq(70, 72, length = 4), sex = "male", nx = 1)
mxf1y <- mlt(seq(78, 79, length = 4), sex = "female", nx = 1)
lcest1y <- lileecarter.estimate(mxm1y, mxf1y, nx = 1)

# projections
pred3 <- mortcast.blend(e0m1y, e0f1y, meth1 = "lc", meth2 = "mlt",
                      weights = c(1,0.25), min.age.groups = 131, nx = 1,
                      meth1.args = list(lc.pars = lcest1y),
                      kannisto.args = list(est.ages = 90:99, proj.ages = 100:130))

# plot results
par(mfrow=c(1,1))
plot(0:130, pred3$female$mx[,5], log = "y", type = "l", col = "red")
lines(0:130, pred3$male$mx[,5], col = "blue")

```

---

pmd

---

*Pattern of Mortality Decline Prediction*


---

## Description

Predict age-specific mortality rates using the Pattern of mortality decline (PMD) method (Andreev et al. 2013).

## Usage

```

pmd(
  e0,
  mx0,
  sex = c("male", "female"),
  nx = 5,
  interp.rho = FALSE,

```

```

    kranges = c(0, 25),
    keep.lt = FALSE,
    keep.rho = FALSE,
    ...
)

modpmd(
  e0,
  mx0,
  sex = c("male", "female"),
  nx = 5,
  interp.rho = FALSE,
  kranges = c(0, 25),
  ax.index = NULL,
  ax.smooth = FALSE,
  ax.smooth.df = NULL,
  keep.lt = FALSE,
  keep.rho = FALSE,
  ...
)

copmd(
  e0m,
  e0f,
  mxm0,
  mxf0,
  nx = 5,
  interp.rho = FALSE,
  keep.rho = FALSE,
  use.modpmd = FALSE,
  ...
)

```

### Arguments

<code>e0</code>	A vector of target life expectancy, one element for each predicted time point.
<code>mx0</code>	A vector with starting age-specific mortality rates. In case of <code>modpmd</code> it can be a matrix where rows correspond to age groups and columns correspond to time periods. Rownames define the starting ages of the age groups.
<code>sex</code>	Either "male" or "female".
<code>nx</code>	Size of age groups. Should be either 5 or 1.
<code>interp.rho</code>	Logical controlling if the $\rho$ coefficients should be interpolated (TRUE) or if the raw (binned) version should be used (FALSE), as stored in the dataset <a href="#">PMDrho</a> .
<code>kranges</code>	A vector of size two, giving the min and max of the $k$ parameter which is estimated to match the target <code>e0</code> using the bisection method.
<code>keep.lt</code>	Logical. If TRUE additional life table columns are kept in the resulting object.
<code>keep.rho</code>	Logical. If TRUE the $\rho$ coefficients are included in the resulting object.

...	Additional arguments passed to the underlying functions. For <code>copmd</code> , in addition to <code>kranges</code> and <code>keep.lt</code> , it can be <code>sexratio.adjust</code> which is a logical controlling if a sex-ratio adjustment should be applied to prevent crossovers between male and female <code>mx</code> . In such a case it uses coefficients from the <code>PMDadjcoef</code> dataset. However, if the argument <code>adjust.with.mxf</code> is set to <code>TRUE</code> (in addition to <code>sexratio.adjust</code> ), the adjustment is done using the female mortality rates as the lower constraint for male mortality rates. If the argument <code>adjust.sr.if.needed</code> is set to <code>TRUE</code> , a sex-ratio adjustment is performed dynamically, using the sex ratio in the previous time point. In such a case, an adjustment in time $t$ is applied only if there was a drop of sex ratio below one at time $t-1$ . Other arguments passed here in <code>copmd</code> can be <code>ax.index</code> , <code>ax.smooth</code> and <code>ax.smooth.df</code> which control the estimation of the initial <code>mx</code> if <code>use.modpmd</code> is <code>TRUE</code> .
<code>ax.index</code>	A vector of column indices of <code>mx</code> to be used to estimate the $a_x = E[\log(mx(t_0))]$ parameter. By default it is estimated as the average over all observed time periods, but this argument can restrict the time periods to use.
<code>ax.smooth</code>	Logical allowing to smooth the $a_x$ over ages.
<code>ax.smooth.df</code>	Degree of freedom for smoothing if <code>ax.smooth</code> is <code>TRUE</code> . Default is half the length of $a_x$ .
<code>e0m</code>	A time series of target male life expectancy.
<code>e0f</code>	A time series of target female life expectancy.
<code>mxm0, mxf0</code>	A vector with starting age-specific male/female mortality rates. If <code>use.modpmd</code> is <code>TRUE</code> , this can be a matrix of historical <code>mx</code> (age x time) from which the starting values are estimated.
<code>use.modpmd</code>	Logical determining if the modified version of PMD ( <code>modpmd</code> ) should be used. In such a case the starting values of mortality rates are estimated similarly to $a_x$ in <code>leecarter.estimate</code> , possibly from more than one time periods. In addition, a smoothing can be applied.

## Details

These functions implements the PMD method introduced in Andreev et al. (2013) and its modifications. It assumes that the future decline in age-specific mortality will follow a certain pattern with the increase in life expectancy at birth ( $e0$ ):

$$\log[mx(t)] = \log[mx(t-1)] - k(t)\rho_x(t)$$

Here,  $\rho_x(t)$  is the age-specific pattern of mortality decline between  $t-1$  and  $t$ . Such patterns for each sex and various levels of  $e0$  are stored in the dataset `PMDrho`. The `pmd` function can be instructed to interpolate between neighboring levels of  $e0$  by setting the argument `interp.rho` to `TRUE`. The  $k$  parameter is estimated to match the  $e0$  level using the bisection method.

Function `pmd` evaluates the method for a single sex, while `copmd` does it coherently for both sexes. In the latter case, the same  $\rho_x$  (namely the average over sex-specific  $\rho_x$ ) is used for both, male and female.

Function `modpmd` implements a modified version of `pmd` where the initial  $\log[mx(t_0)]$  is replaced by an  $a_x$  estimated as in `leecarter.estimate`, i.e. using possibly multiple years of historical `mx` and optionally smoothed. Arguments `ax.index`, `ax.smooth` and `ax.smooth.df` determine the estimation years and parameters of the smoothing.

**Value**

Function `pmd` and `modpmd` return a list with the following elements: a matrix `mx` with the predicted mortality rates. If `keep.lt` is TRUE, it also contains matrices `sr` (survival rates), and life table quantities `Lx` and `lx`. If `keep.rho` is TRUE, it contains a matrix `rho` where columns correspond to the values in the `e0` vector and rows correspond to age groups.

Function `copmd` returns a list with one element for each sex (male and female) where each of them is a list as described above. In addition if `keep.rho` is TRUE, element `rho.sex` gives the sex-dependent (i.e. not averaged)  $\rho_x$  coefficient.

**References**

Andreev, K., Gu, D., Gerland, P. (2013). Age Patterns of Mortality Improvement by Level of Life Expectancy at Birth with Applications to Mortality Projections. Paper presented at the Annual Meeting of the Population Association of America, New Orleans, LA. <https://paa2013.princeton.edu/papers/132554>.

Gu, D., Pelletier, F., Sawyer, C. (2017). Projecting Age-sex-specific Mortality: A Comparison of the Modified Lee-Carter and Pattern of Mortality Decline Methods, UN Population Division, Technical Paper No. 6. New York: United Nations. [https://population.un.org/wpp/Publications/Files/WPP2017\\_TechnicalPaperNo6.pdf](https://population.un.org/wpp/Publications/Files/WPP2017_TechnicalPaperNo6.pdf)

**See Also**

[mortcast](#), [mortcast.blend](#), [PMDrho](#)

**Examples**

```
data(mxF, e0Fproj, package = "wpp2017")
country <- "Hungary"
# get initial mortality for the current year
mxf <- subset(mxF, name == country)[,"2010-2015"]
names(mxf) <- c(0,1, seq(5, 100, by=5))
# get target e0
e0f <- subset(e0Fproj, name == country)[-(1:2)]
# project into future
pred <- pmd(e0f, mxf, sex = "female")
# plot first projection in black and the remaining ones in grey
plot(pred$mx[,1], type = "l", log = "y", ylim = range(pred$mx),
      ylab = "female mx", xlab = "Age", main = country)
for(i in 2:ncol(pred$mx)) lines(pred$mx[,i], col = "grey")
```

**Description**

Data object containing a table of coefficients to be used to adjust the sex ratio in the coherent Pattern Mortality Decline method as implemented in the `copmd` function. To invoke the adjustment, argument `sexratio.adjust` should be set to TRUE.

**Usage**

```
data(PMDadjcoef)
```

**Format**

Data frame containing columns “age”, “intercept”, “l<sub>mxf</sub>”, “e<sub>0f</sub>”, “e<sub>0f2</sub>”, and “gap”. Rows correspond to age groups. The values are estimates of the following regression

$$\log_{10} mx^M = \beta_0 + \beta_1 \log_{10} mx^F + \beta_2 e_0^F + \beta_3 (e_0^F)^2 + \beta_4 (e_0^F - e_0^M)$$

The order of the columns starting with intercept corresponds to the order of the coefficients in the above equation.

**Source**

The coefficients were estimated and provided by Danan Gu, UN Population Division.

**References**

Gu, D., Pelletier, F. and Sawyer, C. (2017). Projecting Age-sex-specific Mortality: A Comparison of the Modified Lee-Carter and Pattern of Mortality Decline Methods, UN Population Division, Technical Paper No. 6. New York: United Nations. [https://population.un.org/wpp/Publications/Files/WPP2017\\_TechnicalPaperNo6.pdf](https://population.un.org/wpp/Publications/Files/WPP2017_TechnicalPaperNo6.pdf)

**See Also**

[copmd](#)

**Examples**

```
data(PMDadjcoef)
PMDadjcoef
```

---

PMDrho

*Pattern Mortality Decline Lookup Tables*

---

**Description**

Data object containing two tables with  $\rho$  coefficients for the Pattern Mortality Decline method as implemented in the [pmd](#) function.

**Usage**

```
data(PMDrho)
```

**Format**

Using `data(PMDrho)` loads two objects into memory: `RhoFemales` and `RhoMales`. They both are data frames with 22 rows corresponding to age groups, and 17 columns corresponding to different levels of life expectancy in 5-years intervals (from 50 to 135). The names of the columns reflect the middle of the respective interval.

**References**

Andreev, K. Gu, D., Gerland, P. (2013). Age Patterns of Mortality Improvement by Level of Life Expectancy at Birth with Applications to Mortality Projections. Paper presented at the Annual Meeting of the Population Association of America, New Orleans, LA. <https://paa2013.princeton.edu/papers/132554>.

Gu, D., Pelletier, F. and Sawyer, C. (2017). Projecting Age-sex-specific Mortality: A Comparison of the Modified Lee-Carter and Pattern of Mortality Decline Methods, UN Population Division, Technical Paper No. 6. New York: United Nations. [https://population.un.org/wpp/Publications/Files/WPP2017\\_TechnicalPaperNo6.pdf](https://population.un.org/wpp/Publications/Files/WPP2017_TechnicalPaperNo6.pdf)

**See Also**

[pmd](#)

**Examples**

```
data(PMDrho)
head(RhoFemales)
head(RhoMales)

# plot a few male patterns
e0lev <- colnames(RhoMales)[c(1, 5, 9, 13, 17)]
plot(RhoMales[, e0lev[1]], type="l", log="y", ylim=range(RhoMales[,e0lev]),
     ylab="male rho", xlab="Age")
for(i in 2:length(e0lev)) lines(RhoMales[,e0lev[i]], lty = i)
legend("bottomleft", legend = e0lev, lty = 1:length(e0lev), bty= "n")
```

---

rotate.leecarter

*Rotated Lee-Carter*

---

**Description**

Rotate the Lee-Carter parameter  $b_x$  over time to reach an ultimate  $b_x^u$ , as described in Li et al. (2013).

**Usage**

```
rotate.leecarter(bx, ultimate.bx, e0, e0l = 80, e0u = 102, p = 0.5)
```

```
ultimate.bx(bx)
```

**Arguments**

bx	A vector of the Lee-Carter $b_x$ parameter (from e.g. <code>lileecarter.estimate</code> or <code>leecarter.estimate</code> ).
ultimate.bx	A vector of the ultimate $b_x^u$ parameter as defined in Li, Lee, Gerland (2013) (obtained using <code>lileecarter.estimate</code> or <code>ultimate.bx</code> ).
e0	A time series of life expectancies.
e0l	Level of life expectancy at which the rotation starts.
e0u	Level of life expectancy at which the rotation finishes.
p	Exponent of the smooth function.

**Value**

Function `rotate.leecarter` returns a matrix of rotated  $B_x(t)$  where rows correspond to age groups and columns correspond to time periods (given by the vector `e0`).

Function `ultimate.bx` returns a vector of the ultimate  $b_x^u$ .

**References**

Li, N., Lee, R. D. and Gerland, P. (2013). Extending the Lee-Carter method to model the rotation of age patterns of mortality decline for long-term projections. *Demography*, 50, 2037-2051.

**Examples**

```
data(mxF, mxM, e0Fproj, e0Mproj, package = "wpp2017")
country <- "Japan"
mxm <- subset(mxM, name == country)[,4:16]
mxf <- subset(mxF, name == country)[,4:16]
e0f <- as.numeric(subset(e0Fproj, name == country)[-(1:2)])
e0m <- as.numeric(subset(e0Mproj, name == country)[-(1:2)])
rownames(mxm) <- rownames(mxf) <- c(0,1, seq(5, 100, by=5))
lc <- lileecarter.estimate(mxm, mxf)
rotlc <- rotate.leecarter(lc$bx, lc$ultimate.bx, (e0f + e0m)/2)
plot(lc$bx, type="l")
lines(lc$ultimate.bx, col="red")
for(i in 1:ncol(rotlc)) lines(rotlc[,i], col="grey")
```

# Index

## \* datasets

LQcoef, 17  
MLTlookup, 19  
PMDadjcoef, 28  
PMDrho, 29

cokannisto, 2, 4, 6, 7, 10, 23, 24  
cokannisto.estimate, 4, 5, 5, 9, 10  
copmd, 23, 24, 28, 29  
copmd (pmd), 25

kannisto, 2, 4, 6, 7, 9  
kannisto.estimate, 7, 8, 9, 10  
kannisto.predict, 4–7, 9, 9

leecarter.estimate, 2, 10, 14, 21, 27, 31  
life.table, 2, 12  
lileecarter.estimate, 2, 11, 14, 21, 31  
logquad, 2, 15, 17, 24  
logquadj, 23  
logquadj (logquad), 15  
LQcoef, 15, 16, 17

mlt, 2, 16, 18, 20  
MLT1Ylookup, 18  
MLT1Ylookup (MLTlookup), 19  
mltj, 23, 24  
mltj (mlt), 18  
MLTlookup, 18, 19, 19  
modpmd (pmd), 25  
MortCast (MortCast-package), 2  
mortcast, 2, 11, 16, 19, 20, 23, 24, 28  
MortCast-package, 2  
mortcast.blend, 2, 16, 19, 21, 22, 28

pmd, 2, 16, 19, 25, 29, 30  
PMDadjcoef, 27, 28  
PMDrho, 26–28, 29

RhoFemales (PMDrho), 29  
RhoMales (PMDrho), 29

rotate.leecarter, 21, 30

ultimate.bx, 31  
ultimate.bx (rotate.leecarter), 30