

# Package ‘ungroup’

December 11, 2019

**Type** Package

**Title** Penalized Composite Link Model for Efficient Estimation of Smooth Distributions from Coarsely Binned Data

**Version** 1.1.5

**Description** Versatile method for ungrouping histograms (binned count data) assuming that counts are Poisson distributed and that the underlying sequence on a fine grid to be estimated is smooth. The method is based on the composite link model and estimation is achieved by maximizing a penalized likelihood. Smooth detailed sequences of counts and rates are so estimated from the binned counts. Ungrouping binned data can be desirable for many reasons: Bins can be too coarse to allow for accurate analysis; comparisons can be hindered when different grouping approaches are used in different histograms; and the last interval is often wide and open-ended and, thus, covers a lot of information in the tail area. Age-at-death distributions grouped in age classes and abridged life tables are examples of binned data. Because of modest assumptions, the approach is suitable for many demographic and epidemiological applications. For a detailed description of the method and applications see Rizzi et al. (2015) <doi:10.1093/aje/kwv020>.

**License** MIT + file LICENSE

**LazyData** TRUE

**Depends** R (>= 3.4.0)

**Imports** MortalitySmooth (>= 2.3.4), pbapply (>= 1.3), Rcpp (>= 0.12.0), rgl (>= 0.99.0), Rdpack (>= 0.8)

**LinkingTo** Rcpp, RcppEigen

**Suggests** MortalityLaws (>= 1.5.0), knitr (>= 1.20), rmarkdown (>= 1.10), testthat (>= 2.0.0)

**RdMacros** Rdpack

**URL** <https://github.com/mpascariu/ungroup>

**BugReports** <https://github.com/mpascariu/ungroup/issues>

**VignetteBuilder** knitr

**RoxygenNote** 6.1.1

**NeedsCompilation** yes

**Author** Marius D. Pascariu [aut, cre] (<<https://orcid.org/0000-0002-2568-6489>>),  
 Silvia Rizzi [aut],  
 Jonas Schoeley [aut] (<<https://orcid.org/0000-0002-3340-8518>>),  
 Maciej J. Danko [aut] (<<https://orcid.org/0000-0002-7924-9022>>)

**Maintainer** Marius D. Pascariu <[rpascariu@outlook.com](mailto:rpascariu@outlook.com)>

**Repository** CRAN

**Date/Publication** 2019-12-11 08:50:03 UTC

## R topics documented:

control.pclm . . . . .	2
control.pclm2D . . . . .	3
pclm . . . . .	4
pclm2D . . . . .	7
plot.pclm . . . . .	9
plot.pclm2D . . . . .	10
residuals.pclm . . . . .	11
residuals.pclm2D . . . . .	12
ungroup . . . . .	13
ungroup.data . . . . .	14

**Index** **16**

---

control.pclm                      *Auxiliary for Controlling pclm Fitting*

---

## Description

Auxiliary for Controlling pclm Fitting

## Usage

```
control.pclm(lambda      = NA,
              kr         = 2,
              deg        = 3,
              int.lambda = c(0.1, 1e+5),
              diff       = 2,
              opt.method = c("BIC", "AIC"),
              max.iter   = 1e+3,
              tol        = 1e-3)
```

**Arguments**

lambda	Smoothing parameter to be used in pclm estimation. If lambda = NA an algorithm will find the optimal values.
kr	Knot ratio. Number of internal intervals used for defining 1 knot in B-spline basis construction. See <a href="#">MortSmooth_bbase</a> .
deg	Degree of the splines needed to create equally-spaced B-splines basis over an abscissa of data.
int.lambda	If lambda is optimized an interval to be searched needs to be specified. Format: vector containing the end-points.
diff	An integer indicating the order of differences of the components of PCLM coefficients. Default value: 2.
opt.method	Selection criterion of the model. Possible values are "AIC" and "BIC". Default: "BIC".
max.iter	Maximal number of iterations used in fitting procedure.
tol	Relative tolerance in PCLM fitting procedure. Default: 0.1% i.e. the estimated aggregate bins should be in the 0.1% error margin.

**Value**

A list with exactly eight control parameters.

**See Also**

[pclm](#)

**Examples**

```
control.pclm()
```

---

```
control.pclm2D
```

*Auxiliary for Controlling pclm2D Fitting*

---

**Description**

Auxiliary for Controlling pclm2D Fitting

**Usage**

```
control.pclm2D(lambda      = c(1, 1),
                kr         = 7,
                deg        = 3,
                int.lambda = c(0.1, 1e+3),
                diff       = 2,
                opt.method = c("BIC", "AIC"),
                max.iter   = 1e+3,
                tol        = 1e-3)
```

**Arguments**

<code>lambda</code>	Smoothing parameter to be used in pclm estimation. If <code>lambda = NA</code> an algorithm will find the optimal values.
<code>kr</code>	Knot ratio. Number of internal intervals used for defining 1 knot in B-spline basis construction. See <a href="#">MortSmooth_bbase</a> .
<code>deg</code>	Degree of the splines needed to create equally-spaced B-splines basis over an abscissa of data.
<code>int.lambda</code>	If <code>lambda</code> is optimized an interval to be searched needs to be specified. Format: vector containing the end-points.
<code>diff</code>	An integer indicating the order of differences of the components of PCLM coefficients. Default value: 2.
<code>opt.method</code>	Selection criterion of the model. Possible values are "AIC" and "BIC". Default: "BIC".
<code>max.iter</code>	Maximal number of iterations used in fitting procedure.
<code>tol</code>	Relative tolerance in PCLM fitting procedure. Default: 0.1% i.e. the estimated aggregate bins should be in the 0.1% error margin.

**Value**

A list with exactly eight control parameters.

**See Also**

[pclm2D](#)

**Examples**

```
control.pclm2D()
```

---

`pclm`

*Univariate Penalized Composite Link Model (PCLM)*

---

**Description**

Fit univariate penalized composite link model (PCLM) to ungroup binned count data, e.g. age-at-death distributions grouped in age classes.

**Usage**

```
pclm(x, y, nlast, offset = NULL, out.step = 1, ci.level = 95,
      verbose = FALSE, control = list())
```

**Arguments**

<code>x</code>	Vector containing the starting values of the input intervals/bins. For example: if we have 3 bins $[0, 5)$ , $[5, 10)$ and $[10, 15)$ , <code>x</code> will be defined by the vector: <code>c(0, 5, 10)</code> .
<code>y</code>	Vector with counts to be ungrouped. It must have the same dimension as <code>x</code> .
<code>nlast</code>	Length of the last interval. In the example above <code>nlast</code> would be 5.
<code>offset</code>	Optional offset term to calculate smooth mortality rates. A vector of the same length as <code>x</code> and <code>y</code> . See Rizzi et al. (2015) for further details.
<code>out.step</code>	Length of estimated intervals in output. Values between 0.1 and 1 are accepted. Default: 1.
<code>ci.level</code>	Level of significance for computing confidence intervals. Default: 95.
<code>verbose</code>	Logical value. Indicates whether a progress bar should be shown or not. Default: FALSE.
<code>control</code>	List with additional parameters: <ul style="list-style-type: none"> <li>• <code>lambda</code> – Smoothing parameter to be used in <code>pclm</code> estimation. If <code>lambda = NA</code> an algorithm will find the optimal values.</li> <li>• <code>kr</code> – Knot ratio. Number of internal intervals used for defining 1 knot in B-spline basis construction. See <a href="#">MortSmooth_bbase</a>.</li> <li>• <code>deg</code> – Degree of the splines needed to create equally-spaced B-splines basis over an abscissa of data.</li> <li>• <code>int.lambda</code> – If <code>lambda</code> is optimized an interval to be searched needs to be specified. Format: vector containing the end-points.</li> <li>• <code>diff</code> – An integer indicating the order of differences of the components of PCLM coefficients.</li> <li>• <code>opt.method</code> – Selection criterion of the model. Possible values are "AIC" and "BIC".</li> <li>• <code>max.iter</code> – Maximal number of iterations used in fitting procedure.</li> <li>• <code>tol</code> – Relative tolerance in PCLM fitting procedure.</li> </ul>

**Details**

The PCLM method is based on the composite link model, which extends standard generalized linear models. It implements the idea that the observed counts, interpreted as realizations from Poisson distributions, are indirect observations of a finer (ungrouped) but latent sequence. This latent sequence represents the distribution of expected means on a fine resolution and has to be estimated from the aggregated data. Estimates are obtained by maximizing a penalized likelihood. This maximization is performed efficiently by a version of the iteratively reweighted least-squares algorithm. Optimal values of the smoothing parameter are chosen by minimizing Bayesian or Akaike's Information Criterion.

**Value**

The output is a list with the following components:

<code>input</code>	A list with arguments provided in <code>input</code> . Saved for convenience.
--------------------	---

fitted	The fitted values of the PCLM model.
ci	Confidence intervals around fitted values.
goodness.of.fit	A list containing goodness of fit measures: standard errors, AIC and BIC.
smoothPar	Estimated smoothing parameters: lambda, kr and deg.
bins.definition	Additional values to identify the bins limits and location in input and output objects.
deep	A list of objects created in the fitting process. Useful in diagnosis of possible issues.
call	An unevaluated function call, that is, an unevaluated expression which consists of the named function applied to the given arguments.

## References

Rizzi S, Gampe J, Eilers PHC (2015). "Efficient Estimation of Smooth Distributions From Coarsely Grouped Data." *American Journal of Epidemiology*, **182**(2), 138-147. doi: [10.1093/aje/kwv020](https://doi.org/10.1093/aje/kwv020).

## See Also

[control.pclmplot.pclm](#)

## Examples

```
# Data
x <- c(0, 1, seq(5, 85, by = 5))
y <- c(294, 66, 32, 44, 170, 284, 287, 293, 361, 600, 998,
      1572, 2529, 4637, 6161, 7369, 10481, 15293, 39016)
offset <- c(114, 440, 509, 492, 628, 618, 576, 580, 634, 657,
           631, 584, 573, 619, 530, 384, 303, 245, 249) * 1000
nlast <- 26 # the size of the last interval

# Example 1 -----
M1 <- pclm(x, y, nlast)
ls(M1)
summary(M1)
fitted(M1)
plot(M1)

# Example 2 -----
# ungroup even in smaller intervals
M2 <- pclm(x, y, nlast, out.step = 0.5)
head(fitted(M1))
plot(M1, type = "s")
# Note, in example 1 we are estimating intervals of length 1. In example 2
# we are estimating intervals of length 0.5 using the same aggregate data.

# Example 3 -----
# Do not optimise smoothing parameters; choose your own. Faster.
M3 <- pclm(x, y, nlast, out.step = 0.5,
```

```

      control = list(lambda = 100, kr = 10, deg = 10))
plot(M3)

summary(M2)
summary(M3) # not the smallest BIC here, but sometimes is not important.

# Example 4 -----
# Grouped x & grouped offset (estimate death rates)
M4 <- pclm(x, y, nlast, offset)
plot(M4, type = "s")

# Example 5 -----
# Grouped x & ungrouped offset (estimate death rates)

ungrouped_Ex <- pclm(x, y = offset, nlast, offset = NULL)$fitted # ungrouped offset data

M5 <- pclm(x, y, nlast, offset = ungrouped_Ex)

```

---

pclm2D

---

*Two-Dimensional Penalized Composite Link Model (PCLM-2D)*


---

## Description

Fit two-dimensional penalized composite link model (PCLM-2D), e.g. simultaneous ungrouping of age-at-death distributions grouped in age classes for adjacent years. The PCLM can be extended to a two-dimensional regression problem. This is particularly suitable for mortality analysis when mortality surfaces are to be estimated to capture both age-specific trajectories of coarsely grouped distributions and time trends (Rizzi et al. 2019).

## Usage

```

pclm2D(x, y, nlast, offset = NULL, out.step = 1, ci.level = 95,
       verbose = TRUE, control = list())

```

## Arguments

x	Vector containing the starting values of the input intervals/bins. For example: if we have 3 bins $[0, 5)$ , $[5, 10)$ and $[10, 15)$ , x will be defined by the vector: $c(0, 5, 10)$ .
y	data.frame with counts to be ungrouped. The number of rows should be equal with the length of x.
nlast	Length of the last interval. In the example above nlast would be 5.
offset	Optional offset term to calculate smooth mortality rates. A vector of the same length as x and y. See Rizzi et al. (2015) for further details.
out.step	Length of estimated intervals in output. Values between 0.1 and 1 are accepted. Default: 1.
ci.level	Level of significance for computing confidence intervals. Default: 95.

verbose	Logical value. Indicates whether a progress bar should be shown or not. Default: TRUE.
control	List with additional parameters: <ul style="list-style-type: none"> <li>• <code>lambda</code> – Smoothing parameter to be used in pclm estimation. If <code>lambda = NA</code> an algorithm will find the optimal values.</li> <li>• <code>kr</code> – Knot ratio. Number of internal intervals used for defining 1 knot in B-spline basis construction. See <a href="#">MortSmooth_bbase</a>.</li> <li>• <code>deg</code> – Degree of the splines needed to create equally-spaced B-splines basis over an abscissa of data.</li> <li>• <code>int.lambda</code> – If <code>lambda</code> is optimized an interval to be searched needs to be specified. Format: vector containing the end-points.</li> <li>• <code>diff</code> – An integer indicating the order of differences of the components of PCLM coefficients.</li> <li>• <code>opt.method</code> – Selection criterion of the model. Possible values are "AIC" and "BIC".</li> <li>• <code>max.iter</code> – Maximal number of iterations used in fitting procedure.</li> <li>• <code>tol</code> – Relative tolerance in PCLM fitting procedure.</li> </ul>

### Value

The output is a list with the following components:

<code>input</code>	A list with arguments provided in input. Saved for convenience.
<code>fitted</code>	The fitted values of the PCLM model.
<code>ci</code>	Confidence intervals around fitted values.
<code>goodness.of.fit</code>	A list containing goodness of fit measures: standard errors, AIC and BIC.
<code>smoothPar</code>	Estimated smoothing parameters: <code>lambda</code> , <code>kr</code> and <code>deg</code> .
<code>bins.definition</code>	Additional values to identify the bins limits and location in input and output objects.
<code>deep</code>	A list of objects created in the fitting process. Useful in diagnosis of possible issues.
<code>call</code>	An unevaluated function call, that is, an unevaluated expression which consists of the named function applied to the given arguments.

### References

Rizzi S, Gampe J, Eilers PHC (2015). "Efficient Estimation of Smooth Distributions From Coarsely Grouped Data." *American Journal of Epidemiology*, **182**(2), 138-147. doi: [10.1093/aje/kwv020](https://doi.org/10.1093/aje/kwv020).

Rizzi S, Halekoh U, Thinggaard M, Engholm G, Christensen N, Johannesen TB, Lindahl-Jacobsen R (2019). "How to estimate mortality trends from grouped vital statistics." *International Journal of Epidemiology*, **48**(2), 571–582. doi: [10.1093/ije/dyy183](https://doi.org/10.1093/ije/dyy183).



**See Also**

[control.pclm2D](#) [plot.pclm2D](#)

**Examples**

```
# Input data
Dx <- ungroup.data$Dx
Ex <- ungroup.data$Ex

# Aggregate data to ungroup it in the examples below
x <- c(0, 1, seq(5, 85, by = 5))
nlast <- 26
n <- c(diff(x), nlast)
group <- rep(x, n)
y <- aggregate(Dx, by = list(group), FUN = "sum")[, -1]
offset <- aggregate(Ex, by = list(group), FUN = "sum")[, -1]

# Example 1 -----
# Fit model and ungroup data using PCLM-2D
P1 <- pclm2D(x, y, nlast)
summary(P1)
# plot(P1)

## Not run:
# NOTE: pclm2D does not search for optimal smoothing parameters by default
# (like pclm) because it is more time consuming. If optimization is required
# set lambda = c(NA, NA):

P1 <- pclm2D(x, y, nlast, control = list(lambda = c(NA, NA)))

# Example 2 -----
# Ungroup and build a mortality surface
P2 <- pclm2D(x, y, nlast, offset)
summary(P2)

plot(P2) # plot
library(rgl)
snapshot3d("plotP2.jpeg") # save the plot in jpeg format
aspect3d(x = 1, y = 2, z = 1) # modify the aspect ratio

## End(Not run)
```

---

plot.pclm

*Generic Plot for pclm Class*

---

**Description**

Generic Plot for pclm Class

**Usage**

```
## S3 method for class 'pclm'
plot(x, xlab, ylab, ylim, type, lwd, col, legend,
     legend.position, ...)
```

**Arguments**

x	An object of class <a href="#">pclm</a>
xlab	a label for the x axis, defaults to a description of x.
ylab	a label for the y axis, defaults to a description of y.
ylim	the y limits of the plot.
type	1-character string giving the type of plot desired. The following values are possible, for details, see <a href="#">plot</a> : "p" for points, "l" for lines, "b" for both points and lines, "c" for empty points joined by lines, "o" for overplotted points and lines, "s" and "S" for stair steps and "h" for histogram-like vertical lines. Finally, "n" does not produce any points or lines.
lwd	Line width, a positive number, defaulting to 2.
col	Three colours to be used in the plot for observed values, fitted values and confidence intervals.
legend	a character or <a href="#">expression</a> vector of length $\geq 1$ to appear in the legend. Other objects will be coerced by <a href="#">as.graphicsAnnot</a> .
legend.position	Legend position, or the x and y co-ordinates to be used to position the legend.
...	other <a href="#">graphical parameters</a> (see <a href="#">par</a> and section 'Details' below).

**See Also**

[pclm](#)

**Examples**

```
# See complete examples in pclm help page
```

---

plot.pclm2D

*Generic Plot for pclm2D Class*

---

**Description**

The generic plot for a `pclm2D` object is constructed using [rgl](#) package. And can be modified/improved using the `rgl` tools implemented in the package like: [surface3d](#), [axes3d](#), [aspect3d](#), [title3d](#) or [snapshot3d](#). For a complete guide to 3D visualization using `rgl` see [this tutorial](#).

**Usage**

```
## S3 method for class 'pclm2D'
plot(x, color = c(1, 2), alpha = c(1, 0.5),
     axes = TRUE, box = TRUE, xlab = "x-axis", ylab = "y-axis",
     zlab = "z-axis", main = "", sub = "", ...)
```

**Arguments**

x	an object of class <a href="#">pclm2D</a>
color	vector of R color characters. Represents the diffuse component in case of lighting calculation (lit = TRUE), otherwise it describes the solid color characteristics.
alpha	vector of alpha values between 0.0 (fully transparent) .. 1.0 (opaque).
axes	add axes to the plot. Logical. Default: TRUE.
box	draw a box around the plot. Logical. Default: TRUE.
xlab	the axis labels for the plot
ylab	the axis labels for the plot
zlab	the axis labels for the plot
main	the main title for the plot
sub	the subtitle for the plot
...	Material and texture properties. See <a href="#">rgl.material</a> for details.

**See Also**

[pclm2D](#) [surface3d](#) [axes3d](#) [aspect3d](#) [title3d](#) [snapshot3d](#)

**Examples**

```
# See complete examples in pclm2D help page
```

---

residuals.pclm

*Extract PCLM Deviance Residuals*


---

**Description**

Extract PCLM Deviance Residuals

**Usage**

```
## S3 method for class 'pclm'
residuals(object, ...)
```

**Arguments**

object            an object for which the extraction of model residuals is meaningful.  
 ...                other arguments.

**Value**

Residuals extracted from the object object.

**Examples**

```
x <- c(0, 1, seq(5, 85, by = 5))
y <- c(294, 66, 32, 44, 170, 284, 287, 293, 361, 600, 998,
      1572, 2529, 4637, 6161, 7369, 10481, 15293, 39016)
M1 <- pclm(x, y, nlast = 26)

residuals(M1)
```

---

 residuals.pclm2D

*Extract PCLM-2D Deviance Residuals*


---

**Description**

Extract PCLM-2D Deviance Residuals

**Usage**

```
## S3 method for class 'pclm2D'
residuals(object, ...)
```

**Arguments**

object            an object for which the extraction of model residuals is meaningful.  
 ...                other arguments.

**Value**

Residuals extracted from the object object.

**Examples**

```
Dx <- ungroup.data$Dx
Ex <- ungroup.data$Ex

# Aggregate data to ungroup it in the example below
x    <- c(0, 1, seq(5, 85, by = 5))
nlast <- 26
n     <- c(diff(x), nlast)
```

```
group <- rep(x, n)
y      <- aggregate(Dx, by = list(group), FUN = "sum")[, -1]

# Example
P1 <- pclm2D(x, y, nlast)

residuals(P1)
```

---

ungroup

*ungroup: Penalized Composite Link Model for Efficient Estimation of Smooth Distributions from Coarsely Binned Data*

---

## Description

Versatile method for ungrouping histograms (binned count data) assuming that counts are Poisson distributed and that the underlying sequence on a fine grid to be estimated is smooth. The method is based on the composite link model and estimation is achieved by maximizing a penalized likelihood. Smooth detailed sequences of counts and rates are so estimated from the binned counts. Ungrouping binned data can be desirable for many reasons: Bins can be too coarse to allow for accurate analysis; comparisons can be hindered when different grouping approaches are used in different histograms; and the last interval is often wide and open-ended and, thus, covers a lot of information in the tail area. Age-at-death distributions grouped in age classes and abridged life tables are examples of binned data. Because of modest assumptions, the approach is suitable for many demographic and epidemiological applications. For a detailed description of the method and applications see Rizzi et al. (2015) <doi:10.1093/aje/kwv020>.

## Details

To learn more about the package, start with the vignettes: `browseVignettes(package = "ungroup")`

## Author(s)

**Maintainer:** Marius D. Pascariu <rpascariu@outlook.com> (0000-0002-2568-6489)

Authors:

- Silvia Rizzi <srizzi@health.sdu.dk>
- Jonas Schoeley (0000-0002-3340-8518)
- Maciej J. Danko <Danko@demogr.mpg.de> (0000-0002-7924-9022)

## References

Currie ID, Durban M, Eilers PH (2004). "Smoothing and forecasting mortality rates." *Statistical modelling*, **4**(4), 279–298.

Eilers PH (2007). "Ill-posed problems with counts, the composite link model and penalized likelihood." *Statistical Modelling*, **7**(3), 239-254. doi: [10.1177/1471082X0700700302](https://doi.org/10.1177/1471082X0700700302).

Hastie TJ, Tibshirani RJ (1990). “Generalized additive models.” *Monographs on Statistics and Applied Probability*, **43**.

Human Mortality Database (2018). “University of California, Berkeley (USA), and Max Planck Institute for Demographic Research (Germany). Data downloaded on 17/01/2018.” <https://www.mortality.org>.

Pascariu MD (2018). *MortalityLaws: Parametric Mortality Models, Life Tables and HMD*. R package version 1.6.0, <https://github.com/mpascariu/MortalityLaws>.

Rizzi S, Gampe J, Eilers PHC (2015). “Efficient Estimation of Smooth Distributions From Coarsely Grouped Data.” *American Journal of Epidemiology*, **182**(2), 138-147. doi: [10.1093/aje/kwv020](https://doi.org/10.1093/aje/kwv020).

Rizzi S, Halekoh U, Thinggaard M, Engholm G, Christensen N, Johannesen TB, Lindahl-Jacobsen R (2019). “How to estimate mortality trends from grouped vital statistics.” *International Journal of Epidemiology*, **48**(2), 571–582. doi: [10.1093/ije/dyy183](https://doi.org/10.1093/ije/dyy183).

Rizzi S, Thinggaard M, Engholm G, Christensen N, Johannesen TB, Vaupel JW, Lindahl-Jacobsen R (2016). “Comparison of non-parametric methods for ungrouping coarsely aggregated data.” *BMC medical research methodology*, **16**(1), 59. doi: [10.1186/s1287401601578](https://doi.org/10.1186/s1287401601578).

Thompson R, Baker R (1981). “Composite link functions in generalized linear models.” *Applied Statistics*, 125–131.

## See Also

Useful links:

- <https://github.com/mpascariu/ungroup>
- Report bugs at <https://github.com/mpascariu/ungroup/issues>

---

ungroup.data

*Test Dataset in the Package*

---

## Description

Dataset containing death counts (D<sub>x</sub>) and exposures (E<sub>x</sub>) by age for a certain population between 1980 and 2014. The data-set is provided for testing purposes only and might be altered and outdated. Download actual demographic data free of charge from Human Mortality Database (2018). Once a username and a password is created on the [website](#) the **MortalityLaws** R package can be used to extract data in R format.

## Usage

ungroup.data

### **Format**

An object of class `ungroup.data` of length 2.

### **Source**

[Human Mortality Database](#)

### **References**

Human Mortality Database (2018). “University of California, Berkeley (USA), and Max Planck Institute for Demographic Research (Germany). Data downloaded on 17/01/2018.” <https://www.mortality.org>.

Pascariu MD (2018). *MortalityLaws: Parametric Mortality Models, Life Tables and HMD*. R package version 1.6.0, <https://github.com/mpascariu/MortalityLaws>.

### **See Also**

[ReadHMD](#)

# Index

## \*Topic **datasets**

- ungroup.data, [14](#)
  
- as.graphicsAnnot, [10](#)
- aspect3d, [10](#), [11](#)
- axes3d, [10](#), [11](#)
  
- control.pclm, [2](#), [6](#)
- control.pclm2D, [3](#), [9](#)
  
- expression, [10](#)
  
- graphical parameters, [10](#)
  
- MortSmooth\_bbase, [3–5](#), [8](#)
  
- par, [10](#)
- pclm, [3](#), [4](#), [10](#)
- pclm2D, [4](#), [7](#), [11](#)
- plot, [10](#)
- plot.pclm, [6](#), [9](#)
- plot.pclm2D, [9](#), [10](#)
  
- ReadHMD, [15](#)
- residuals.pclm, [11](#)
- residuals.pclm2D, [12](#)
- rgl, [10](#)
- rgl.material, [11](#)
  
- snapshot3d, [10](#), [11](#)
- surface3d, [10](#), [11](#)
  
- title3d, [10](#), [11](#)
  
- ungroup, [13](#)
- ungroup-package (ungroup), [13](#)
- ungroup.data, [14](#)