Package 'rwlifetable'

May 9, 2026

Type Package

Title Estimation of Life Tables Using Rolling Windows

Version 0.1.0-6

Maintainer Jose M. Pavía <jose.m.pavia@uv.es>

Description Estimates life tables, specifically (crude) death rates and (raw and graduated) death probabilities, using rolling windows in one (e.g., age), two (e.g., age and time) or three (e.g., age, time and income) dimensions. The package can also be utilised for summarising statistics and smoothing continuous variables through rolling windows in other domains, such as estimating averages of self-positioning ideology in political science.
Acknowledgements:
The authors wish to thank Ministerio de Ciencia, Innovación y Universidades (grant PID2021-128228NB-I00) and Generalitat Valenciana (grants HIECPU/2023/2, Conselleria de Hacienda, Economía y Administración Pública, and CIGE/2023/7, Conselleria de Educación, Cultura, Universidades y Empleo) for supporting this research.

License EPL

Encoding UTF-8

Imports stats, splines

RoxygenNote 7.2.3

NeedsCompilation no

Author Jose M. Pavía [aut, cre] (ORCID: <https://orcid.org/0000-0002-0129-726X>), Josep Lledó [aut] (ORCID: <https://orcid.org/0000-0002-7475-8549>)

Repository CRAN

Date/Publication 2025-01-16 11:00:06 UTC

Contents

dw_crude_mx 2
dw_preprocess 4
graduate_qx 5
tw_crude_mx 9

Index 13

dw_crude_mx *Computes crude/raw death rates/probabilities using double rolling windows*

Description

This function estimates (crude/raw) death rates/probabilities using rolling windows in two dimensions (usually, age and time) from a given experience of mortality, summarized in a dataset of (either initial or times of) exposed to risk and a dataset of recorded deaths.

Usage

```
dw_crude_mx(
  exposed,
  deaths,
  age.window = 2,
  time.window = 2,
  age.wb = age.window,
  age.wf = age.window,
  time.wb = time.window,
  time.wf = time.window,
  weights = 1,
  initial = FALSE,
  partial = FALSE
)
```

Arguments

exposed	A matrix (or data frame) of order nxT (where n is the number of ages and T the number of years), with ages by rows and years by columns (with no column for ages). For each age-year combination, the matrix contains either the initial number of individuals exposed to risk in the portfolio (population) or the time of exposition to risk of the exposed in the portfolio (population). By default, the matrix contains time exposures. When 'exposed' contains initial number of individuals exposed to risk, time exposures are estimated assuming uniformity and a closed demographic system. It is assumed that the order of rows and columns in 'deaths' and 'exposed' is consistent (i.e., identical), and that both ages and years are sequential.
deaths	A matrix (or data frame) of order nxT, with ages by rows and years by columns (with no column for ages), containing the number of deaths recorded in the portfolio (population), where n is the number of ages and T the number of years. It is assumed that the order of rows and columns in 'deaths' and 'exposed' is consistent (i.e., are the same), and that both ages and years are sequential.
age.window	An non-negative integer indicating the number of preceding and succeeding ages to be included in the age rolling window. Default, 2.

`time.window` An non-negative integer indicating the number of preceding and succeeding years (time periods) to be included in the year rolling window. Default, 2.

`age.wb` An non-negative integer indicating the number of preceding ages to be included in the age rolling window. Default, 'age.window'. If 'age.wb' and 'age.window' differ, the definition of 'age.wb' takes precedence, allowing non-centered age windows.

`age.wf` An non-negative integer indicating the number of succeeding ages to be included in the age rolling window. Default, 'age.window'. If 'age.wf' and 'age.window' differ, the definition of 'age.wf' takes precedence, allowing non-centered age windows.

`time.wb` An non-negative integer indicating the number of preceding years to be included in the year rolling window. Default, 'time.window'. If 'time.wb' and 'time.window' differ, the definition of 'time.wb' takes precedence, allowing non-centered time windows.

`time.wf` An non-negative integer indicating the number of succeeding years to be included in the year rolling window. Default, 'time.window'. If 'time.wf' and 'time.window' differ, the definition of 'time.wf' takes precedence, allowing non-centered time windows.

`weights` A matrix of order (age.wb + age.wf + 1) x (time.wb + time.wf + 1) of real positive numbers indicating the multiplicative factors to be applied to the different components of the double rolling window. Default, '1'; meaning all the components of the rolling window are aggregated as available in 'exposed' and 'deaths'.

`initial` A TRUE/FALSE argument indicating whether the numbers in 'exposed' represent the initial number of individuals exposed to risk in the portfolio (population), which corresponds to 'initial = TRUE', or the time exposed to risk, with 'initial = FALSE' as default.

`partial` A TRUE/FALSE argument indicating whether estimates obtained using incomplete rolling windows should be also computed. Default, 'FALSE'; meaning crude death rates are estimated only for combinations of (year, age) where the corresponding double rolling window is complete.

Value

A list with six matrices: 'mx', 'qx', 'Lx', 'dx', 'Lx.total', 'dx.total'. When 'partial = FALSE' these matrices are of order (n - age.wb - age.wf) x (T - time.wb - time.wf); otherwise, they are of order nxT. The names of the rows and of the columns of these matrices are inherited from the corresponding names in the 'deaths' object, ensuring that if the row names indicate ages and the column names indicate years, this contextual information is preserved.

`mx` The 'mx' matrix holds the crude death rates estimated after determining the number of individuals (time) exposed to risk and deaths to be used for each estimate using the specified rolling windows with the defined 'weights'.

`qx` The 'qx' matrix contains the raw death probabilities, derived from 'mx', based on the assumption that, on average, each deceased individual lives for half a year in the year of their death.

Lx	The ‘Lx‘ matrix contains the actual time exposures used to compute the estimates.
dx	The ‘dx‘ matrix contains the number of deaths used to compute the estimates..
Lx.total	The ‘Lx.total‘ matrix contains the total numbers for time exposures which corresponds to all factor-weights being equal. If ‘weights‘ is a constant matrix ‘Lx‘ and ‘Lx.total‘ coincide.
dx.total	The ‘dx.total‘ matrix contains the total numbers deaths which corresponds to all factor-weights being equal. If ‘weights‘ is a constant matrix ‘dx‘ and ‘dx.total‘ coincide.

Note

The function could be used to apply double rolling windows to datasets from other fields other than demography and actuarial science.

Author(s)

Jose M. Pavia <pavia@uv.es>

Josep Lledo <josep.lledo@uv.es>

Examples

```
exposed <- structure(list(Year2017 = c(6078.14, 5841.78, 5575.70, 5726.18, 5458.21, 5197.56,
                                       5018.12, 4791.56, 4245.15, 4321.65, 4179.3),
                          Year2018 = c(5978.73, 5473.78, 5572.23, 5495.19, 5148.47, 4845.14,
                                       4739.54, 4222.01, 4476.99, 4306.45, 4108.58),
                          Year2019 = c(5593.23, 5551.41, 5260.44, 5079.56, 4873.37, 4857.78,
                                       4536.12, 4453.85, 4310.89, 4015.02, 3974.25)),
                              class = "data.frame", row.names = 68:78)

deaths <- structure(list(Year2017 = c(144, 102, 113, 122, 156, 110, 126, 132, 120, 172, 110),
                         Year2018 = c(111, 122, 109, 116, 162, 154, 115, 146, 100, 169, 146),
                         Year2019 = c(100, 123, 113, 151, 122, 110, 137, 175, 137, 110, 155)),
                            class = "data.frame", row.names = 68:78)

example <- dw_crude_mx(exposed = exposed, deaths = deaths,
                       age.window = 2, time.window = 1, initial = FALSE)
```

---

dw_preprocess *Pre-processes a data frame of deaths, exposed to risk or times of exposition to be used by the ‘dw_crude_mx‘ function*

Description

This function pre-processes a data frame (or matrix) of deaths, exposed to risk or times of exposition to be used by the ‘dw_crude_mx‘ function when the data frame has a variable of ages in the first column and likely data are non-available for all the sequence of ages desired by the user. It creates a new data frame by inserting zeroes in the cells corresponding to the missing ages.

Usage

```
dw_preprocess(x, age.min = 0, age.max = 120)
```

Arguments

x	A data frame (or matrix) of deaths, exposed to risk or times of exposition with ages as first column.
age.min	A non-negative integer indicating the minimum age for the output object. Default, 0.
age.max	A non-negative integer indicating the maximum age for the output object. Default, 120.

Value

A data.frame containing the same columns as ‘x‘ except for the first one, as outlined in the function's description.

Author(s)

Jose M. Pavia <pavia@uv.es>

Examples

```
deaths <- structure(list(ages = c(67, 69:78),
                      Year2017 = c(144, 102, 113, 122, 156, 110, 126, 132, 120, 172, 110),
                      Year2018 = c(111, 122, 109, 116, 162, 154, 115, 146, 100, 169, 146),
                     Year2019 = c(100, 123, 113, 151, 122, 110, 137, 175, 137, 110, 155)),
                         class = "data.frame", row.names = 1:11)

example <- dw_preprocess(x = deaths)
```

---

graduate_qx *Graduates raw death probabilities*

Description

This function graduates raw death probabilities using the two-step approach proposed by Sifre-Armengol et al. (2024). The method involves (i) applying natural cubic splines to smooth the log-transformation of the raw estimates, and (ii) correcting unexpected decreases in mortality trends at the oldest ages by modelling and extending death probability estimates up to a higher maximum age (omega). The function accepts as main input either an output from ‘dw_crude_mx‘ or ‘tw_crude_mx‘, or a list with containing similar ‘qx‘, ‘Lx‘ and ‘dx‘ components.

Usage

```
graduate_qx(
  x,
  ages,
  df = max(5, round(range(ages)/5)),
  threshold = 75,
  size = 10,
  omega = 120,
  zero2NA = 0,
  max.c.NA = 3,
  plot.explore = FALSE,
  ...
)
```

Arguments

x	A list that is either an output of the ‘dw_crude_mx‘ or ‘tw_crude_mx‘ functions, or a list containing similar ‘qx‘, ‘Lx‘ and ‘dx‘ components as the outputs of these functions. It is assumed that the first dimension of the matrices/arrays in ‘x‘ corresponds to age. Likewise, as happens with ‘dw_crude_mx‘ or ‘tw_crude_mx‘ is assumed that ages are ordered sequentially.
ages	A vector of two components of non-negative integers indicating the minimum and maximum ages of the matrices in ‘x‘.
df	A non-negative integer indicating the number of degrees of freedom to be used in the natural splines. Default, the greater between 5 and the range of ‘ages‘ divided by 5. It is recommended to test several values for ‘df‘ if the smooth does not look appropriate.
threshold	A non-negative integer indicating the minimum age from which start to explore whether unexpected decreases in mortality trends are observed. Default, 75.
size	A non-negative integer indicating the sample size to be used for modelling (default is 10). No modelling is performed for a series of estimates if, after determining the effective threshold, the number of available observations is insufficient.
omega	A non-negative integer indicating the initial maximum age up to which death probabilities are to be extended. By default, this is set to the maximum between ‘omega‘ (default value 120) and the maximum age in ‘x‘.
zero2NA	A nonnegative number to be used as a threshold for excluding particularly small raw estimated death probabilities. Default, ‘0‘.
max.c.NA	A nonnegative number informing about the maximum consequetive NA allowed in the range selected for smoothing. Default, ‘3‘.
plot.explore	A TRUE/FALSE argument indicating whether exploratory figures with the outputs should be plotted as a side-effect. Default, ‘FALSE‘.
...	Additional parameters to be passed to the ‘ns‘ function, from the ‘spline‘ package, as this is function is used to smooth the raw probabilities.

Value

A list with three components (qx.raw, qx.smooth, qx.final) with the same dimensions than qx. The first component (qx.raw) contains the introduced raw death probabilities. The second component (qx.smooth) contains the smoothed death probabilities obtained after applying the natural splines. The third (qx.final) component contains the final smoothed death probabilities attained after modelling, when possible and required.

Note

When a set of final smoothed estimates cannot be obtained due to unsuccessful modeling, all corresponding values within the output component qx.final are set to 'NA'.

Author(s)

Jose M. Pavia <pavia@uv.es>

References

Sifre-Armengol, C., Pavia J.M. and Lledo, J. (2025) A comprehensive database of estimates and forecasts of Spanish sex-age death rates by climate area, income level, and habitat size (2010-2050). *Demographic Research*, 52(1): 1-24. doi:10.4054/DemRes.2025.52.1.

Examples

```
qx <- matrix(c(0.00053, 1e-04, 0.00013, 1e-04, 9e-05, 6e-05, 4e-05, 3e-05,
               4e-05, 9e-05, 8e-05, 0.00017, 0.00025, 0.00031, 0.00029,
               0.00034, 0.00035, 0.00035, 0.00036, 0.00034, 0.00035, 0.00034,
               0.00037, 0.00037, 0.00045, 0.00048, 5e-04, 0.00056, 0.00068,
               1e-04, 0.00073, 0.00077, 0.00073, 0.00074, 0.00081, 0.00086,
               0.00099, 0.00107, 0.00111, 0.00116, 0.00122, 0.00142, 0.00167,
               0.00197, 0.00235, 0.00282, 0.00305, 0.00327, 0.00357, 0.0039,
               0.00438, 0.00481, 0.00571, 0.00615, 0.00676, 0.00728, 0.00787,
               0.00839, 0.0094, 0.01044, 0.01119, 0.01231, 0.01336, 0.01456,
               0.01551, 0.0168, 0.01834, 0.02012, 0.0214, 0.02341, 0.02565,
               0.02744, 0.02995, 0.03275, 0.03604, 0.04027, 0.04617, 0.05092,
               0.05793, 0.06425, 0.07136, 0.07907, 0.0878, 0.09703, 0.10735,
               0.11913, 0.13115, 0.14478, 0.15927, 0.17506, 0.19133, 0.20803,
               0.22459, 0.23739, 0.25289, 0.26508, 0.26514, 0.27186, 0.28433,
               0.27071, 0.23445, 0.20622, 0.17334, 0.13063, 0.12384, 0.1298,
               0.14223, 0.10746, 0.1071, 0.09006, 0.08212, 0.00044, 0.00015,
               0.00011, 0.00014, 0.00011, 6e-05, 3e-05, 4e-05, 5e-05, 0.00011,
               1e-04, 0.00016, 0.00021, 0.00027, 0.00028, 0.00029, 0.00035,
               0.00036, 0.00037, 0.00034, 0.00038, 0.00036, 0.00038, 0.00037,
               0.00044, 5e-04, 0.00051, 0.00055, 7e-04, 0.00074, 0.00076, 0.00078,
               0.00076, 7e-04, 0.00078, 0.00078, 0.00091, 0.00096, 0.00103,
               0.00112, 0.0012, 0.00138, 0.0016, 0.00192, 0.00224, 0.00266,
               0.00291, 0.00319, 0.00349, 0.00387, 0.0044, 0.00491, 0.00565,
               0.00609, 0.00673, 0.00723, 0.00778, 0.00844, 0.00952, 0.01044,
               0.01119, 0.0124, 0.01357, 0.01488, 0.01594, 0.01727, 0.01872,
               0.02029, 0.02171, 0.0237, 0.02625, 0.02818, 0.03051, 0.03323,
               0.03698, 0.04079, 0.04605, 0.05167, 0.05898, 0.06439, 0.07144,
```

```
                0.07935, 0.08765, 0.09585, 0.10583, 0.11705, 0.12861, 0.14218,
                0.15571, 0.17194, 0.18712, 0.20318, 0.21773, 0.2308, 0.24627,
                0.25896, 0.26325, 0.26827, 0.2765, 0.27272, 0.24799, 0.22936,
                0.19868, 0.17734, 0.15662, 0.15714, 0.14942, 0.12339, 0.14107,
                0.08829, 0.06772),
                nrow = 111L, ncol = 2L, dimnames = list(NULL, c("Y2021", "Y2022")))

Lx <- matrix(c(39493.05, 48856.36, 55123.06, 60751.17, 66031.82, 70946.31,
                75208.69, 79412.74, 82652.24, 85128.87, 87020.23, 88685.67,
                89705, 90517.94, 91646.01, 92918.48, 94170.36, 95460.83, 97687.18,
                100164.66, 103261.96, 106992.27, 111921.92, 116603.89, 121291.46,
                125936.06, 130752.09, 134804.76, 139320.39, 144206.5, 149682.01,
                154873.73, 160815.58, 167089.22, 173680.55, 179504.54, 184932.72,
                189330.61, 192820.72, 193978.13, 193934.13, 193179.32, 191721.7,
                188989.66, 186417.68, 184378.77, 182201.46, 180054.42, 178485.35,
                177261.95, 175410.36, 173603.62, 171631.32, 169582.45, 167628.59,
                165317.96, 162698.1, 159899.55, 156710.46, 152477.14, 148809.7,
                144888.18, 141481.5, 138641.49, 136507.83, 133596.61, 131455.18,
                129085.66, 125392.97, 120104.96, 114700.89, 107965.81, 100031.66,
                91822.18, 84574.49, 78886.03, 74100.05, 70091.95, 67604.06, 65699.18,
                62590.7, 58770.4, 54806.82, 49668.38, 43748.81, 37750.22, 31925.7,
                26140.64, 21029.01, 16690.26, 12842.08, 9656.27, 7150.16, 5130.6,
                3537.15, 2395.44, 1599.79, 1058.39, 675.82, 447.16, 308.76, 234.85,
                179.15, 157.42, 136.35, 122.47, 97.96, 88.06, 79.53, 63.62, 46.71,
                36630.78, 45891.8, 52172.94, 57352.35, 62562.67, 67386.39, 72005.75,
                76355.92, 80515.9, 83898.66, 86538.01, 88545.23, 90128.12, 91180.86,
                92148.96, 93368.73, 94695.83, 95926.17, 97595.46, 99320.7, 101814.36,
                104643.36, 108574.15, 112812.37, 117584.14, 121903.48, 126651.53,
                130739.43, 134924.59, 139100.46, 144051.46, 148808.85, 154075.99,
                159886.92, 166350.24, 172505.02, 178361.41, 183831.91, 188630.54,
                191534.35, 193004.78, 193429.58, 193222.93, 191243.25, 188992.12,
                186776.36, 184924.44, 182272.74, 180435.57, 179322.72, 177992.62,
                176083.51, 174604.92, 172917.28, 170794.39, 168626.14, 166345.52,
                163790.65, 160928.87, 157002.29, 152973.22, 149050.82, 144971.78,
                141000.87, 138484.3, 135568.46, 132464.84, 129908.86, 127041.89,
                122810.99, 117770.62, 111814.78, 105165.19, 97433.74, 89018.38,
                81824.61, 76121.28, 70589.37, 66375.32, 63833.38, 61269.47, 57992.61,
                54404.69, 50190.06, 45060.81, 39460.55, 33700.22, 27923.89, 22693.17,
                17989.68, 13907.67, 10479.36, 7768.35, 5603.56, 3927.38, 2645.58,
                1771.36, 1171.63, 751.12, 468.68, 307.32, 227.74, 167.73, 138.75,
                123.58, 105.55, 92.89, 68.44, 59.3, 54.13, 42.8),
                nrow = 111L, ncol = 2L, dimnames = list(NULL, c("Y2021", "Y2022")))

dx <- matrix(c(21, 5, 7, 6, 6, 4, 3, 2, 3, 8, 7, 15, 22, 28, 27, 32, 33, 33,
                35, 34, 36, 36, 41, 43, 55, 61, 66, 75, 95, 101, 110, 119,
                118, 124, 141, 154, 183, 203, 215, 225, 237, 275, 321, 373,
                438, 520, 557, 589, 639, 693, 770, 837, 982, 1047, 1137, 1208,
                1285, 1347, 1480, 1600, 1674, 1794, 1903, 2034, 2134, 2263,
                2433, 2623, 2712, 2845, 2980, 3004, 3041, 3057, 3104, 3242, 3502,
                3662, 4033, 4361, 4632, 4838, 5033, 5065, 4963, 4782, 4481, 4080,
                3639, 3202, 2717, 2242, 1809, 1382, 1024, 732, 489, 333, 224,
                140, 82, 54, 34, 22, 18, 17, 15, 10, 9, 6, 4, 16, 7, 6, 8, 7,
                4, 2, 3, 4, 9, 9, 14, 19, 25, 26, 27, 33, 35, 36, 34, 39, 38,
```

```
                41, 42, 52, 61, 64, 72, 95, 103, 110, 116, 117, 112, 129, 134,
                162, 177, 194, 214, 231, 268, 310, 367, 424, 498, 539, 583, 631,
                696, 785, 867, 990, 1056, 1153, 1224, 1300, 1389, 1539, 1647,
                1722, 1860, 1981, 2114, 2225, 2361, 2503, 2663, 2788, 2946, 3132,
                3196, 3258, 3292, 3354, 3407, 3588, 3744, 4034, 4247, 4539, 4792,
                4987, 5053, 5035, 4906, 4632, 4274, 3832, 3384, 2871, 2370, 1898,
                1462, 1103, 787, 537, 363, 241, 148, 87, 59, 37, 27, 21, 18,
                15, 9, 9, 5, 3),
                nrow = 111L, ncol = 2L, dimnames = list(NULL, c("Y2021", "Y2022")))

x <- list("qx" = qx, "Lx" = Lx, "dx" = dx)

example <- graduate_qx(x = x, ages = c(3, 113))
```

`tw_crude_mx` *Computes crude/raw death rates/probabilities using triple rolling windows*

Description

This function estimates (crude/raw) death rates/probabilities (life tables) using rolling windows in three dimensions (e.g., age, time and income) from a given experience of mortality, summarized in a three-dimension array of (either initial or times of) exposed to risk and a three-dimension dataset of recorded deaths.

Usage

```
tw_crude_mx(
  exposed,
  deaths,
  dim1.window = 2,
  dim2.window = 2,
  dim3.window = 2,
  dim1.wb = dim1.window,
  dim1.wf = dim1.window,
  dim2.wb = dim2.window,
  dim2.wf = dim2.window,
  dim3.wb = dim3.window,
  dim3.wf = dim3.window,
  weights = 1,
  initial = FALSE,
  partial = FALSE
)
```

Arguments

`exposed`	A three-dimensional array of order n×T×I. It is recommended (and when 'initial = TRUE' assumed) that the first dimension corresponds to age. For each

row-column-slice combination, the array contains either the initial number of individuals exposed to risk in the portfolio (population) or the exposure time for those at risk within the portfolio. By default, the array contains exposure times. When 'exposed' contains the initial number of individuals exposed to risk, (i) the first dimension represents must represent ages, and (ii) exposure times are estimated based on the assumptions of uniformity and a closed demographic system. It is also assumed that the order of rows, columns, and slices in 'deaths' and 'exposed' is consistent (i.e., identical) and that the sequence of values (levels) across the three dimensions is ordered sequentially.

`deaths` A three-dimension array of order nxTxI. It is recommended (and when 'initial = TRUE' assumed) that the first dimension corresponds to age. For each combination of row-column-slice, the array contains the number of deaths recorded in the portfolio (population). It is assumed that the order of rows, columns, and slices in 'deaths' and 'exposed' is consistent (i.e., identical) and that the sequence of values (levels) across the three dimensions is ordered sequentially.

`dim1.window` An non-negative integer indicating the number of preceding and succeeding levels to be included in the rolling window for dimension 1. Default, 2.

`dim2.window` An non-negative integer indicating the number of preceding and succeeding levels to be included in the rolling window for dimension 2. Default, 2.

`dim3.window` An non-negative integer indicating the number of preceding and succeeding levels to be included in the rolling window for dimension 3. Default, 2.

`dim1.wb` An non-negative integer indicating the number of preceding levels to be included in the rolling window for dimension 1. Default, 'dim1.window'. If 'dim1.wb' and 'dim1.window' differ, the definition of 'dim1.wb' takes precedence, allowing non-centered dimension 1 windows.

`dim1.wf` An non-negative integer indicating the number of succeeding levels to be included in the rolling window for dimension 1. Default, 'dim1.window'. If 'dim1.wf' and 'dim1.window' differ, the definition of 'dim1.wf' takes precedence, allowing non-centered dimension 1 windows.

`dim2.wb` An non-negative integer indicating the number of preceding levels to be included in the rolling window for dimension 2. Default, 'dim2.window'. If 'dim2.wb' and 'dim2.window' differ, the definition of 'dim2.wb' takes precedence, allowing non-centered dimension 2 windows.

`dim2.wf` An non-negative integer indicating the number of succeeding levels to be included in the rolling window for dimension 2. Default, 'dim2.window'. If 'dim2.wf' and 'dim2.window' differ, the definition of 'dim2.wf' takes precedence, allowing non-centered dimension 2 windows.

`dim3.wb` An non-negative integer indicating the number of preceding levels to be included in the rolling window for dimension 3. Default, 'dim3.window'. If 'dim3.wb' and 'dim3.window' differ, the definition of 'dim3.wb' takes precedence, allowing non-centered dimension 3 windows.

`dim3.wf` An non-negative integer indicating the number of succeeding levels to be included in the rolling window for dimension 3. Default, 'dim3.window'. If 'dim3.wf' and 'dim3.window' differ, the definition of 'dim3.wf' takes precedence, allowing non-centered dimension 3 windows.

`weights` A three-dimension array of order (dim1.wb + dim1.wf + 1) x (dim2.wb + dim2.wf + 1) x (dim3.wb + dim3.wf + 1) of real positive numbers indicating the multiplicative factors to be applied to the different components of the triple rolling window. Default, '1'; meaning all the components of the rolling window are aggregated as available in 'exposed' and 'deaths'.

`initial` A TRUE/FALSE argument indicating whether the numbers in 'exposed' represent the initial number of individuals exposed to risk in the portfolio (population), which corresponds to 'initial = TRUE', or the time exposed to risk, with 'initial = FALSE' as default.

`partial` A TRUE/FALSE argument indicating whether estimates obtained using incomplete rolling windows should be also computed. Default, 'FALSE'; meaning crude death rates are estimated only for combinations where the corresponding triple rolling window is complete.

Value

A list with six three-dimension arrays: 'mx', 'qx', 'Lx', 'dx', 'Lx.total', 'dx.total'. When 'partial = FALSE' these arrays are of order (n - dim1.wb - dim1.wf) x (T - dim2.wb - dim2.wf) x (I - dim3.wb - dim3.wf); otherwise, they are of order nxTxI. The names of the rows, columns and slices of these arrays are inherited from the corresponding names in the 'deaths' object, ensuring that if the row, column and slice names indicate relevant data, this contextual information is preserved.

`mx` The 'mx' array holds the crude death rates estimated after determining the number of individuals (time) exposed to risk and deaths to be used for each estimate using the specified rolling windows with the defined 'weights'.

`qx` The 'qx' array contains the raw death probabilities, derived from 'mx', based on the assumption that, on average, each deceased individual lives for half a year in the year of their death.

`Lx` The 'Lx' array contains the actual time exposures used to compute the estimates.

`dx` The 'dx' array contains the number of deaths used to compute the estimates..

`Lx.total` The 'Lx.total' array contains the total numbers for time exposures which corresponds to all factor-weights being equal. If 'weights' is a constant matrix 'Lx' and 'Lx.total' coincide.

`dx.total` The 'dx.total' array contains the total numbers deaths which corresponds to all factor-weights being equal. If 'weights' is a constant array 'dx' and 'dx.total' coincide.

Note

The first dimension in the objects 'exposed' and 'deaths' should correspond to the variable age when the object 'exposed' represents the initial number of individuals exposed to risk in the portfolio (population). The function could be used to apply triple rolling windows to datasets from other fields other than demography and actuarial science.

Author(s)

Jose M. Pavia <pavia@uv.es>

Josep Lledo <josep.lledo@uv.es>

Examples

```
exposed0 <- structure(list(Year2017 = c(6078.14, 5841.78, 5575.70, 5726.18, 5458.21, 5197.56,
                                        5018.12, 4791.56, 4245.15, 4321.65, 4179.3),
                           Year2018 = c(5978.73, 5473.78, 5572.23, 5495.19, 5148.47, 4845.14,
                                        4739.54, 4222.01, 4476.99, 4306.45, 4108.58),
                           Year2019 = c(5593.23, 5551.41, 5260.44, 5079.56, 4873.37, 4857.78,
                                        4536.12, 4453.85, 4310.89, 4015.02, 3974.25)),
                                class = "data.frame", row.names = 68:78)


deaths0 <- structure(list(Year2017 = c(144, 102, 113, 122, 156, 110, 126, 132, 120, 172, 110),
                          Year2018 = c(111, 122, 109, 116, 162, 154, 115, 146, 100, 169, 146),
                          Year2019 = c(100, 123, 113, 151, 122, 110, 137, 175, 137, 110, 155)),
                              class = "data.frame", row.names = 68:78)

exposed <- deaths <- array(NA, dim = c(11, 3, 5))
set.seed(123)
for (kk in 1:5){
   exposed[, , kk] <- as.matrix(exposed0) +
                         matrix(runif(33, 0, 100), nrow = 11, ncol = 3)
   deaths[, , kk] <- as.matrix(deaths0) +
                         matrix(runif(33, 0, 5), nrow = 11, ncol = 3)
}
dimnames(exposed) <- dimnames(deaths) <- list(68:78,
                                              paste0("Y", 2017:2019),
                                              paste0("I", 1:5))

example <- tw_crude_mx(exposed = exposed, deaths = deaths, dim1.window = 2,
                       dim2.window = 1, dim3.window = 1, initial = FALSE)
```

Index

dw_crude_mx, 2
dw_preprocess, 4

graduate_qx, 5

tw_crude_mx, 9