

# Package ‘mem’

July 25, 2025

**Version** 2.19

**Date** 2025-07-25

**Title** The Moving Epidemic Method

**Description** The Moving Epidemic Method, created by T Vega and JE Lozano (2012, 2015) <[doi:10.1111/j.1750-2659.2012.00422.x](https://doi.org/10.1111/j.1750-2659.2012.00422.x)>, <[doi:10.1111/irv.12330](https://doi.org/10.1111/irv.12330)>, allows the weekly assessment of the epidemic and intensity status to help in routine respiratory infections surveillance in health systems. Allows the comparison of different epidemic indicators, timing and shape with past epidemics and across different regions or countries with different surveillance systems. Also, it gives a measure of the performance of the method in terms of sensitivity and specificity of the alert week.

**Depends** R (>= 3.4.0)

**Imports** sm, boot, RColorBrewer, mclust, ggplot2, tidyr, dplyr, purrr, RcppRoll, EnvStats, methods

**Suggests** magick

**License** GPL (>= 2)

**URL** <https://github.com/lozalojo/mem>

**BugReports** <https://github.com/lozalojo/mem/issues>

**Encoding** UTF-8

**RoxygenNote** 7.3.2

**NeedsCompilation** no

**Author** Jose E. Lozano [aut, cre]

**Maintainer** Jose E. Lozano <[lozalojo@gmail.com](mailto:lozalojo@gmail.com)>

**Repository** CRAN

**Date/Publication** 2025-07-25 09:10:01 UTC

## Contents

|                     |   |
|---------------------|---|
| epimem . . . . .    | 2 |
| epitiming . . . . . | 3 |

|                          |           |
|--------------------------|-----------|
| flucyl                   | 4         |
| flucylraw                | 5         |
| full.series.graph        | 6         |
| memevolution             | 8         |
| memgoodness              | 10        |
| memintensity             | 14        |
| memmodel                 | 15        |
| memstability             | 20        |
| memsurveillance          | 21        |
| memsurveillance.animated | 24        |
| memtiming                | 26        |
| memtrend                 | 28        |
| optimum.by.inspection    | 30        |
| processPlots             | 32        |
| roc.analysis             | 33        |
| transformdata            | 35        |
| transformdata.back       | 37        |
| transformseries          | 38        |
| <b>Index</b>             | <b>41</b> |

---

|        |                                                  |
|--------|--------------------------------------------------|
| epimem | <i>Deprecated function(s) in the mem package</i> |
|--------|--------------------------------------------------|

---

**Description**

These functions are provided for compatibility with older version of the mem package. They may eventually be completely removed.

**Usage**

```
epimem(  
  i.data,  
  i.tails = 1,  
  i.levels = c(0.4, 0.9, 0.975),  
  i.type = 2,  
  i.level = 0.95,  
  ...  
)
```

**Arguments**

|          |                                                  |
|----------|--------------------------------------------------|
| i.data   | Original parameter deprecated in the new version |
| i.tails  | Original parameter deprecated in the new version |
| i.levels | Original parameter deprecated in the new version |
| i.type   | Original parameter deprecated in the new version |
| i.level  | Original parameter deprecated in the new version |

...

Parameters (that have not changed) to be passed to the modern version of the function

Details

epimem

now a synonym for [memmodel](#)

---

|           |                                                  |
|-----------|--------------------------------------------------|
| epitiming | <i>Deprecated function(s) in the mem package</i> |
|-----------|--------------------------------------------------|

---

Description

These functions are provided for compatibility with older version of the mem package. They may eventually be completely removed.

Usage

```
epitiming(i.data, ...)
```

Arguments

|        |                                                                                       |
|--------|---------------------------------------------------------------------------------------|
| i.data | Original parameter deprecated in the new version                                      |
| ...    | Parameters (that have not changed) to be passed to the modern version of the function |

Details

epitiming

now a synonym for [mentiming](#)

flucyl

Castilla y Leon influenza crude rates

## Description

This data set contains *Influenza Like Illness* (ILI) rates, in cases per 100,000 inhabitants collected by the *Influenza Surveillance Programme* of the **Castilla y Leon Health Sentinel Network** (CyLHSN) from 2001 to 2008.

## Usage

```
data(flucyl)
```

## Format

A data frame with 33 observations on 8 variables. Each observation is one surveillance week, and each variable is an influenza season.

2001/2002 a numeric vector - 2001/2002 rates per 100,000 inhabitants.  
 2002/2003 a numeric vector - 2002/2003 rates per 100,000 inhabitants.  
 2003/2004 a numeric vector - 2003/2004 rates per 100,000 inhabitants.  
 2004/2005 a numeric vector - 2004/2005 rates per 100,000 inhabitants.  
 2005/2006 a numeric vector - 2005/2006 rates per 100,000 inhabitants.  
 2006/2007 a numeric vector - 2006/2007 rates per 100,000 inhabitants.  
 2007/2008 a numeric vector - 2007/2008 rates per 100,000 inhabitants.  
 2008/2009 a numeric vector - 2008/2009 rates per 100,000 inhabitants.

## Details

The **Castilla y Leon Health Sentinel Network** is a spanish regional influenza surveillance system based upon volunteer health professionals. The *Influenza Surveillance Programme* consists on a random sample of general practitioners (covering 30,000 population) which collect ILI cases weekly from 40th week (October) to 20th week (May) of the following year to provide estimations of the ILI weekly rate for the entire region.

The data set contains 8 surveillance seasons, from 2001/2002 to 2008/2009.

## Source

Influenza Surveillance Programme. Castilla y Leon Health Sentinel Network. Consejeria de Sanidad. Junta de Castilla y Leon 2001-2008.

## References

Castilla y Leon Health Sentinel Network Reports (Informes de la Red Centinela Sanitaria de Castilla y Leon).  
 Influenza Surveillance Programme (Programa de vigilancia de la gripe).  
<https://www.saludcastillayleon.es/profesionales/es/centinelas>

## Examples

```
data(flucyl)
plot(flucyl[, 1], type = "l")
```

---

flucylraw

*Castilla y Leon influenza standarised rates*


---

## Description

This data set contains *Influenza Like Illness* (ILI) rates, in cases per 100,000 inhabitants collected by the *Influenza Surveillance Programme* of the **Castilla y Leon Health Sentinel Network** (CyLHSN) from 2001 to 2008.

## Usage

```
data(flucylraw)
```

## Format

A data frame with 267 observations on 2 variables. Each observation is one surveillance week and rate,

year a numeric vector - year.

week a numeric vector - week.

rates a numeric vector - standarised rates per 100,000 inhabitants.

## Details

The **Castilla y Leon Health Sentinel Network** is a spanish regional influenza surveillance system based upon volunteer health professionals. The *Influenza Surveillance Programme* consists on a random sample of general practitioners (covering 30,000 population) which collect ILI cases weekly from 40th week (October) to 20th week (May) of the following year to provide estimations of the ILI weekly rate for the entire region.

The data set contains 8 surveillance seasons, from 2001/2002 to 2008/2009.

## Source

Influenza Surveillance Programme. Castilla y Leon Health Sentinel Network. Consejeria de Sanidad. Junta de Castilla y Leon 2001-2008.

## References

Castilla y Leon Health Sentinel Network Reports (Informes de la Red Centinela Sanitaria de Castilla y Leon).

Influenza Surveillance Programme (Programa de vigilancia de la gripe).

<https://www.saludcastillayleon.es/profesionales/es/centinelas>

**Examples**

```
data(flucylraw)
plot(flucylraw$rates, type = "l")
```

---

|                   |                                                            |
|-------------------|------------------------------------------------------------|
| full.series.graph | <i>Creates the historical series graph of the datasets</i> |
|-------------------|------------------------------------------------------------|

---

**Description**

Function full.series.graph creates a graph with the whole dataset.

**Usage**

```
full.series.graph(
  i.data,
  i.range.x = NA,
  i.range.y = NA,
  i.output = ".",
  i.graph.title = "",
  i.graph.subtitle = "",
  i.graph.file = TRUE,
  i.graph.file.name = "",
  i.plot.timing = FALSE,
  i.plot.intensity = FALSE,
  i.alternative.thresholds = NA,
  i.color.pattern = c("#C0C0C0", "#606060", "#000000", "#808080", "#000000", "#001933",
    "#00C000", "#800080", "#FFB401", "#8c6bb1", "#88419d", "#810f7c", "#4d004b"),
  i.mem.info = TRUE,
  ...
)
```

**Arguments**

|                   |                                        |
|-------------------|----------------------------------------|
| i.data            | Historical data series.                |
| i.range.x         | Range x (surveillance weeks) of graph. |
| i.range.y         | Range y of graph.                      |
| i.output          | Directory where graph is saved.        |
| i.graph.title     | Title of the graph.                    |
| i.graph.subtitle  | Subtitle of the graph.                 |
| i.graph.file      | Graph to a file.                       |
| i.graph.file.name | Name of the graph.                     |
| i.plot.timing     | Plot the timing of epidemics.          |

|                          |                                                                                                                |
|--------------------------|----------------------------------------------------------------------------------------------------------------|
| i.plot.intensity         | Plot the intensity levels.                                                                                     |
| i.alternative.thresholds | Use alternative thresholds, instead of the ones modelled by the input data (epidemic + 3 intensity thresholds) |
| i.color.pattern          | colors to use in the graph.                                                                                    |
| i.mem.info               | include information about the package in the graph.                                                            |
| ...                      | other parameters passed to memmodel.                                                                           |

### Details

Input data must be a data.frame with each column a surveillance season and each row a week.

The resulting graph is a time series-like plot showing all the columns in the original dataset one after another.

Color codes:

1. Axis.
2. Tickmarks.
3. Axis labels.
4. Series line.
5. Series dots (default).
6. Title and subtitle.
7. Series dots (pre-epidemic).
8. Series dots (epidemic).
9. Series dots (post-epidemic).
10. Epidemic threshold.
11. Medium threshold.
12. High threshold.
13. Very high threshold.

### Value

full.series.graph writes a tiff graph of the full series of the dataset.

### Author(s)

Jose E. Lozano <lozalojo@gmail.com>

## References

Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? Int Congr Ser. 2004 Jun;1263:281-3.

Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. Influenza Other Respir Viruses. 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.

Vega T, Lozano JE, Meerhoff T, Snacken R, Beaute J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. Influenza Other Respir Viruses. 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.

Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983

## Examples

```
# Castilla y Leon Influenza Rates data
data(flucyl)
# Data of the last season
# uncomment to execute
# full.series.graph(flucyl)
```

---

memevolution

*Evolution of estimators*


---

## Description

Function memeolution analyzes the evolution of mem estimators

## Usage

```
memevolution(
  i.data,
  i.evolution.seasons = 10,
  i.evolution.method = "sequential",
  ...
)
```

## Arguments

|                                  |                                      |
|----------------------------------|--------------------------------------|
| <code>i.data</code>              | Data frame of input data.            |
| <code>i.evolution.seasons</code> | Maximum number of seasons to use.    |
| <code>i.evolution.method</code>  | method to calculate evolution.       |
| <code>...</code>                 | other parameters passed to memmodel. |

## Details

Shows the evolution of some indicators across time. The indicators are: duration of the epidemic, start of the epidemic, epidemic percentage, pre-epidemic, post-epidemic and intensity thresholds.

Values per season are the estimates of the indicator for this specific season calculated according the [memmodel](#) options: cross or sequential (validation option), maximum number of seasons to use in calculations. In [memmodel](#) is it also possible to select the method used to calculate indicators and the level of confidence of the confidence intervals.

To clarify how it is calculated, the `evolution.seasons` shows for each x-axis point the seasons selected to calculate thresholds. Each row is a different season and in each column it is shown if that specific season has been used to calculate the indicators. If TRUE, it has been used, if NO, it hasn't. In the last row, called Next, the indicators are calculated for the upcoming next season.

Note that if you select sequential validation, first points in the graphs calculate thresholds using less data (there are less seasons before the current data than the max. seasons value), and this will affect the range of confidence intervals.

There are four indicators:

**Duration** Average duration of the epidemic and its confidence interval.

**Start** Average start of the epidemic and its confidence interval.

**Epidemic percentage** The sum of the values (cases/rates) in the epidemic period divided by the total sum of values of the whole surveillance period. It's a coverage percentage of the epidemic period. And its confidence interval.

**Thresholds** Pre-epidemic, post-epidemic and intensity thresholds.

The confidence intervals for these indicators are calculated using the other CI and other CI. level parameters of [memmodel](#).

## Value

`memevolution` returns a list. A list containing at least the following components:

**evolution.data** data for each value analysed.

## Author(s)

Jose E. Lozano <lozalojo@gmail.com>

## References

Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? *Int Congr Ser.* 2004 Jun;1263:281-3.

Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. *Influenza Other Respir Viruses.* 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.

Vega T, Lozano JE, Meerhoff T, Snacken R, Beaute J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. *Influenza Other Respir Viruses.* 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.

Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983

## Examples

```
# Castilla y Leon Influenza Rates data
data(flucyl)
# evolution of estimators
evolution <- memevolution(flucyl)
evolution$evolution.data
```

---

memgoodness

*Goodness of fit of the mem*


---

## Description

Function memgoodness calculates different indicators related to the goodness of the MEM for detecting the epidemics, using data from the model and using all data in the original dataset.

## Usage

```
memgoodness(
  i.data,
  i.seasons = 10,
  i.type.threshold = 5,
  i.level.threshold = 0.95,
  i.tails.threshold = 1,
  i.type.intensity = 6,
  i.level.intensity = c(0.4, 0.9, 0.975),
  i.tails.intensity = 1,
  i.type.curve = 2,
  i.level.curve = 0.95,
  i.type.other = 2,
  i.level.other = 0.95,
  i.method = 2,
  i.param = 2.8,
  i.n.max = -1,
  i.type.boot = "norm",
  i.iter.boot = 10000,
  i.calculation.method = "default",
  i.goodness.method = "cross",
  i.goodness.threshold = NA,
  i.goodness.intensity = NA,
  i.detection.values = seq(1, 5, 0.1),
  i.weeks.above = 1,
  i.output = ".",
  i.graph = FALSE,
  i.prefix = "",
  i.min.seasons = 6,
  i.labels.axis = c("Week", "Weekly rate"),
  i.labels.periods = c("Pre", "Epidemic", "Post"),
```

```

i.labels.intensities = c("Epidemic thr", "Medium thr", "High thr", "Very high thr"),
i.labels.details = c("algorithm", "threshold", "Method used",
  "weeks above/below the threshold", "week(s) above the threshold", "Sensitivity",
  "Specificity")
)

```

## Arguments

|                                   |                                                                                |
|-----------------------------------|--------------------------------------------------------------------------------|
| <code>i.data</code>               | Data frame of input data.                                                      |
| <code>i.seasons</code>            | Maximum number of seasons to use.                                              |
| <code>i.type.threshold</code>     | Type of confidence interval to calculate the threshold.                        |
| <code>i.level.threshold</code>    | Level of confidence interval to calculate the threshold.                       |
| <code>i.tails.threshold</code>    | Tails for the confidence interval to calculate the threshold.                  |
| <code>i.type.intensity</code>     | Type of confidence interval to calculate the intensity thresholds.             |
| <code>i.level.intensity</code>    | Levels of confidence interval to calculate the intensity thresholds.           |
| <code>i.tails.intensity</code>    | Tails for the confidence interval to calculate the threshold.                  |
| <code>i.type.curve</code>         | Type of confidence interval to calculate the modelled curve.                   |
| <code>i.level.curve</code>        | Level of confidence interval to calculate the modelled curve.                  |
| <code>i.type.other</code>         | Type of confidence interval to calculate length, start and percentages.        |
| <code>i.level.other</code>        | Level of confidence interval to calculate length, start and percentages.       |
| <code>i.method</code>             | Method to calculate the optimal timing of the epidemic.                        |
| <code>i.param</code>              | Parameter to calculate the optimal timing of the epidemic.                     |
| <code>i.n.max</code>              | Number of pre-epidemic values used to calculate the threshold.                 |
| <code>i.type.boot</code>          | Type of bootstrap technique.                                                   |
| <code>i.iter.boot</code>          | Number of bootstrap iterations.                                                |
| <code>i.calculation.method</code> | method of determining true/false positives and true/false negatives.           |
| <code>i.goodness.method</code>    | method to calculate goodness.                                                  |
| <code>i.goodness.threshold</code> | pre/post epidemic thresholds for threshold goodness method.                    |
| <code>i.goodness.intensity</code> | intensitie thresholds (medium, high, very high) for threshold goodness method. |
| <code>i.detection.values</code>   | values to use in the <code>i.param</code> value of <code>memtiming</code> .    |
| <code>i.weeks.above</code>        | number of weeks over the threshold to give the alert.                          |
| <code>i.output</code>             | output directory for graphs.                                                   |

|                                   |                                                           |
|-----------------------------------|-----------------------------------------------------------|
| <code>i.graph</code>              | whether the graphs must be written or not.                |
| <code>i.prefix</code>             | prefix used for naming graphs.                            |
| <code>i.min.seasons</code>        | minimum number of seasons to perform goodness, default=6. |
| <code>i.labels.axis</code>        | different labels used by output graphs                    |
| <code>i.labels.periods</code>     | different labels used by output graphs                    |
| <code>i.labels.intensities</code> | different labels used by output graphs                    |
| <code>i.labels.details</code>     | different labels used by output graphs                    |

## Details

The indicators calculated are sensitivity, specificity, positive predictive value, negative predictive value, percent agreement and the Matthews correlation coefficient.

How goodness is calculated:

MEM calculates goodness indicators in an iterative process. In each iteration:

1. For one particular season the timing is calculated to determine which weeks are inside the pre, post and epidemic periods. This is used as the real data: a real positive outcome (epidemic weeks) and a real negative outcome (pre and post-epidemic weeks).
2. With a set of seasons, pre-epidemic threshold is calculated. This threshold is compared with values from the season selected in the first step and see if values are above or below the threshold. This is used as the observed data: an observed positive outcome (week value above the threshold), and observed negative outcome (week value below the threshold).
3. Each week has a real and an observed outcome, so it can be classified in:
  - True positives (TP)** real positive, observed positive: values of the epidemic period above the threshold.
  - True negatives (TN)** real negative, observed negative: values of the non-epidemic period below the threshold.
  - False positives (FP)** real negative, observed positive: values of the non-epidemic period above the threshold.
  - False negatives (FN)** real positive, observed negative: values of the epidemic period below the threshold.
4. The process is repeated for each season in the dataset (each iteration a different value until all seasons have been processed).
5. All TP, TN, FP and FN are pooled together and sensitivity, specificity, positive predictive value, negative predictive value, percent agreement and the Matthews correlation coefficient are calculated.

There are two ways of deciding the set of seasons used to calculate the pre-epidemic threshold in each iteration and it is determined by the `i.goodness.method`.

**cross** For each value, the surrounding seasons (after or before the current value) are selected up to the number of Max. seasons (parameter of the Model box). To calculate the thresholds for season 2010/2011, data from 2005/2006 to 2009/2010 and from 2011/20012 to 2015/2016 will be taken.

**sequential** Only preceding seasons are used (before the current value) up to the number of Max. seasons. To calculate the thresholds for season 2010/2011, data from 2000/2001 to 2009/2010 are taken.

**threshold** The pre/post epidemic and intensity thresholds are fixed values for all the seasons and are compared with the epidemic as determined by MEM algorithm.

The `i.calculation.method` is used to determine when the alert based on the epidemic threshold. The "default" method sets all values above the threshold as epidemic, and all the values below as non-epidemic. Pre-epidemic threshold is used for values before the peak, and post-epidemic threshold for values after the peak. The "alternative" method sets an epidemic start and epidemic end, all the values in between are epidemic values. The epidemic start is when `i.weeks.above` (default=1) consecutive weeks are above the pre-epidemic threshold. The epidemic end is the first week below the post-epidemic threshold after the epidemic start. Note that if no post-epidemic threshold is provided, the pre-epidemic value is used instead.

## Value

`memgoodness` returns a list. A list containing at least the following components:

**validity.data** data for each value analysed.

**results** Total weeks, non-missing weeks, true positives, false positives true negatives, false negatives, sensitivity, specificity .

**peaks** distribution of the levels of intensity of the peaks.

**peaks.data** Peak value, week of the peak value, epidemic and intensity thresholds and intensity level of each season analysed.

## Author(s)

Jose E. Lozano <lozalojo@gmail.com>

## References

Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? Int Congr Ser. 2004 Jun;1263:281-3.

Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. Influenza Other Respir Viruses. 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.

Vega T, Lozano JE, Meerhoff T, Snacken R, Beaute J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. Influenza Other Respir Viruses. 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.

Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983

## Examples

```
# Castilla y Leon Influenza Rates data
data(flucyl)
# Goodness of fit
```

```

epi.good <- memgoodness(flucyl, i.detection.values = seq(2.6, 2.8, 0.1))
epi.good$results
epi.good$peaks

```

---

|              |                                           |
|--------------|-------------------------------------------|
| memintensity | <i>Thresholds for influenza intensity</i> |
|--------------|-------------------------------------------|

---

## Description

Function memintensity is used to calculate the thresholds for influenza activity using historical records (surveillance rates).

## Usage

```
memintensity(i.flu)
```

## Arguments

`i.flu`                      An object of class mem.

## Details

This method is based on the Moving Epidemics Method (MEM) used to monitor influenza activity in a weekly surveillance system.

Input data is a data frame containing rates that represent historical influenza surveillance data. It can start and end at any given week (typically at week 40th), and rates can be expressed as per 100,000 inhabitants (or per consultations, if population is not available) or any other scale.

MEM is used to locate the epidemic for each season. Then confidence intervals are calculated at different levels.

The parameter `i.levels`, define the three levels of the confidence intervals used to calculate thresholds.

The `i.n.max` parameter indicates how many epidemic values to use to calculate the thresholds. A value of -1 indicates the program to use an appropriate number of points depending on the number of seasons provided as input.

The `i.seasons` parameter indicates how many seasons are used for calculating thresholds. A value of -1 indicates the program to use as many as possible. If there are less than this parameter, the program used all seasons available.

Intensity thresholds and Epidemic threshold defines 5 levels of intensity:

1. Baseline level - Below epidemic threshold.
2. Low level - Above epidemic threshold and below low intensity threshold.
3. Medium level - Above low intensity threshold and below medium intensity threshold.
4. High level - Above medium intensity threshold and below high intensity threshold.
5. Very high level - Above high intensity threshold.

**Value**

memintensity returns a list with three objects, two of them are the parameters used (param.levels and param.seasons) and the third one (intensity.thresholds) is a matrix 1x4 with the epidemic and intensity thresholds.

1. Epidemic threshold.
2. Low intensity threshold.
3. Medium intensity threshold.
4. High intensity threshold.

**Author(s)**

Jose E. Lozano <lozalojo@gmail.com>

**References**

Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? Int Congr Ser. 2004 Jun;1263:281-3.

Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. Influenza Other Respir Viruses. 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.

Vega T, Lozano JE, Meerhoff T, Snacken R, Beaute J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. Influenza Other Respir Viruses. 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.

Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983

**Examples**

```
# Castilla y Leon Influenza Rates data
data(flucyl)
# mem model
flucyl.mem <- memmodel(flucyl)
# Calculates intensity thresholds
intensity <- memintensity(flucyl.mem)
intensity
```

**Description**

Function memmodel is used to calculate the threshold for influenza epidemic using historical records (surveillance rates).

The method to calculate the threshold is described in the Moving Epidemics Method (MEM) used to monitor influenza activity in a weekly surveillance system.

**Usage**

```
memmodel(
  i.data,
  i.seasons = 10,
  i.type.threshold = 5,
  i.level.threshold = 0.95,
  i.tails.threshold = 1,
  i.type.intensity = 6,
  i.level.intensity = c(0.4, 0.9, 0.975),
  i.tails.intensity = 1,
  i.type.curve = 2,
  i.level.curve = 0.95,
  i.type.other = 3,
  i.level.other = 0.95,
  i.method = 2,
  i.param = 2.8,
  i.centering = -1,
  i.n.max = -1,
  i.type.boot = "norm",
  i.iter.boot = 10000,
  i.mem.info = TRUE,
  i.use.t = FALSE
)
```

**Arguments**

|                                |                                                                          |
|--------------------------------|--------------------------------------------------------------------------|
| <code>i.data</code>            | Data frame of input data.                                                |
| <code>i.seasons</code>         | Maximum number of seasons to use.                                        |
| <code>i.type.threshold</code>  | Type of confidence interval to calculate the threshold.                  |
| <code>i.level.threshold</code> | Level of confidence interval to calculate the threshold.                 |
| <code>i.tails.threshold</code> | Tails for the confidence interval to calculate the threshold.            |
| <code>i.type.intensity</code>  | Type of confidence interval to calculate the intensity thresholds.       |
| <code>i.level.intensity</code> | Levels of confidence interval to calculate the intensity thresholds.     |
| <code>i.tails.intensity</code> | Tails for the confidence interval to calculate the threshold.            |
| <code>i.type.curve</code>      | Type of confidence interval to calculate the modelled curve.             |
| <code>i.level.curve</code>     | Level of confidence interval to calculate the modelled curve.            |
| <code>i.type.other</code>      | Type of confidence interval to calculate length, start and percentages.  |
| <code>i.level.other</code>     | Level of confidence interval to calculate length, start and percentages. |
| <code>i.method</code>          | Method to calculate the optimal timing of the epidemic.                  |

|                          |                                                                                     |
|--------------------------|-------------------------------------------------------------------------------------|
| <code>i.param</code>     | Parameter to calculate the optimal timing of the epidemic.                          |
| <code>i.centering</code> | Number of weeks to center the moving seasons.                                       |
| <code>i.n.max</code>     | Number of pre-epidemic values used to calculate the threshold.                      |
| <code>i.type.boot</code> | Type of bootstrap technique.                                                        |
| <code>i.iter.boot</code> | Number of bootstrap iterations.                                                     |
| <code>i.mem.info</code>  | include information about the package in the graph.                                 |
| <code>i.use.t</code>     | Whether to use t-values (t distribution) instead of z-values (normal distribution). |

## Details

Input data is a data frame containing rates that represent historical influenza surveillance data. It can start and end at any given week (typically at week 40th), and rates can be expressed as per 100,000 inhabitants (or per consultations, if population is not available) or any other scale.

Parameters `i.type`, `i.type.threshold` and `i.type.curve` defines how to calculate confidence intervals along the process.

`i.type.curve` is used for calculating the typical influenza curve, `i.type.threshold` is used to calculate the pre and post epidemic threshold and `i.type` is used for any other confidence interval used in the method.

All three parameters must be a number between 1 and 6:

- 1 Arithmetic mean and mean confidence interval.
- 2 Geometric mean and mean confidence interval.
- 3 Median and the Nyblom/normal approximation confidence interval.
- 4 Median and bootstrap confidence interval.
- 5 Arithmetic mean and point confidence interval (standard deviations).
- 6 Geometric mean and point confidence interval (standard deviations).

Option 3 uses the Hettmansperger and Sheather (1986) and Nyblom (1992) method, when there is enough sample size. If sample size is small, then the normal approximation will be used as described in Conover, 1980, p. 112. Refer to EnvStats package for more information.

Option 4 uses two more parameters: `i.type.boot` indicates which bootstrap method to use. The values are the same of those of the `boot::boot.ci` function. Parameter `i.iter.boot` indicates the number of bootstrap samples to use. See `boot::boot` for more information about this topic.

Parameters `i.level`, `i.level.threshold` and `i.level.curve` indicates, respectively, the level of the confidence intervals described above.

The `i.n.max` parameter indicates how many pre epidemic values to use to calculate the threshold. A value of -1 indicates the program to use an appropriate number of points depending on the number of seasons provided as input. `i.tails` tells the program to use 1 or 2 tailed confidence intervals when calculating the threshold (1 is recommended).

Parameters `i.method` and `i.param` indicates how to find the optimal timing of the epidemics. See [mentiming](#) for details on the values this parameters can have.

The `i.use.t` parameter allows to use the t-value (from a t distribution) instead of the z-value (from a normal distribution) for confidence intervals that rely on z/t-values. Useful when using less than 30 values to calculate confidence intervals.

It is important to know how to arrange information in order to use with memapp. The key points are:

1. One single epidemic wave each season.
2. Never delete a rate inside an epidemic.
3. Accommodate week 53.
4. Do not inflate missing values with zeroes.

Data must contain information from the historical series. Surveillance period can start and end at any given week (typically start at week 40th and ends at week 20th), and data can have any units and can be expressed in any scale (typically rates per 100,000 inhabitants or consultations).

The table must have one row per epidemiological week and one column per surveillance season. A season is a full surveillance period from the beginning to the end, where occurs at some point one single epidemic wave on it. No epidemic wave can be spared in two consecutive seasons. If so, you have to redefine the start and end of the season defined in your dataset. If a season have two waves, it must be split in two periods and must be named accordingly with the seasons name conventions described below. Each cell contains the value for a given week in a given season.

The first column should contain the names of the weeks. When the season contains two different calendar years, the week will go from 40th of the first year to 52nd, and then from 1st to 20th. When the season contains one year, the weeks will go from 1st to 52nd.

Note: If there is no column with week names, the application will name the weeks numbering from 1 to the number of rows.

1. In the northern hemisphere countries, the surveillance period usually goes from week 40 to 20 of the following year (notation: season 2016/2017).
2. In the southern hemisphere countries, the surveillance period usually goes from week 18 to 39 same year (notation: season 2017).

The first row must contain the names of the seasons. This application understand the naming of a season when it contains one or two four digits year separated by / and one one-digit number between parenthesis to identify the wave number. The wave number part in a name of a season is used when a single surveillance period has two epidemic waves that have to be separated in order to have reliable results. In this case, each wave is placed in different columns and named ending with (1) for the first period, (2) for the second, and so on.

## Value

memmodel returns an object of class mem. An object of class mem is a list containing at least the following components:

**i.data** input data

**pre.post.intervals** Pre/post confidence intervals (Threshold is the upper limit of the confidence interval).

**ci.length** Mean epidemic length confidence interval.

**ci.percent** Mean covered percentage confidence interval.

**mean.length** Mean length.

**moving.epidemics** Moving epidemic rates.  
**mean.start** Mean epidemic start.  
**epi.intervals** Epidemic levels of intensity.  
**typ.curve** Typical epidemic curve.  
**n.max** Effective number of pre epidemic values.

## Author(s)

Jose E. Lozano <lozalojo@gmail.com>

## References

- Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? Int Congr Ser. 2004 Jun;1263:281-3.
- Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. Influenza Other Respir Viruses. 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.
- Vega T, Lozano JE, Meerhoff T, Snacken R, Beaute J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. Influenza Other Respir Viruses. 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.
- Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983
- Hettmansperger, T. P., and S. J. Sheather. 1986. Confidence Intervals Based on Interpolated Order Statistics. Statistics and Probability Letters 4: 75-79. doi:10.1016/0167-7152(86)90021-0.
- Nyblom, J. 1992. Note on Interpolated Order Statistics. Statistics and Probability Letters 14: 129-31. doi:10.1016/0167-7152(92)90076-H.
- Conover, W.J. (1980). Practical Nonparametric Statistics. Second Edition. John Wiley and Sons, New York.

## Examples

```
# Castilla y Leon Influenza Rates data
data(flucyl)
# Finds the timing of the first season: 2001/2002
epi <- memmodel(flucyl)
print(epi)
summary(epi)
plot(epi)
```

---

|              |                                |
|--------------|--------------------------------|
| memstability | <i>Stability of indicators</i> |
|--------------|--------------------------------|

---

## Description

Function memstability perform an analisis of stability of estimators

## Usage

```
memstability(i.data, ...)
```

## Arguments

|        |                                      |
|--------|--------------------------------------|
| i.data | Data frame of input data.            |
| ...    | other parameters passed to memmodel. |

## Details

Shows the stability of the main indicators according the number of seasons used to calculate them. That allows to decide the minimum and maximum number of seasons to be included in the model for this specific country and parameter.

These indicators are calculated for the next upcoming season, so for all the iterations are used the last number of seasons determined by the x-axis, so that 2 means 'last two seasons', 10 means 'last ten seasons' and so on.

The `stability.seasons` output shows data used to calculate each particular point in the graph.

## Value

memstability returns a list. A list containing at least the following components:

**stability.data** data for each value analysed.

**stability.seasons** seasons used for each iteration.

## Author(s)

Jose E. Lozano <lozalojo@gmail.com>

## References

Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? Int Congr Ser. 2004 Jun;1263:281-3.

Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. Influenza Other Respir Viruses. 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.

Vega T, Lozano JE, Meerhoff T, Snacken R, Beaute J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. Influenza Other Respir Viruses. 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.

Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983

## Examples

```
# Castilla y Leon Influenza Rates data
data(flucyl)
# Stability
stability <- memstability(flucyl)
stability$stability.data
```

---

|                 |                                                             |
|-----------------|-------------------------------------------------------------|
| memsurveillance | <i>Creates the surveillance graph of the current season</i> |
|-----------------|-------------------------------------------------------------|

---

## Description

Function memsurveillance creates a surveillance graph for the current season.

## Usage

```
memsurveillance(
  i.current,
  i.epidemic.thresholds = NA,
  i.intensity.thresholds = NA,
  i.mean.length = 10,
  i.force.length = FALSE,
  i.output = ".",
  i.graph.title = "",
  i.graph.subtitle = "",
  i.graph.file = TRUE,
  i.graph.file.name = "",
  i.week.report = NA,
  i.equal = FALSE,
  i.pos.epidemic = FALSE,
  i.no.epidemic = FALSE,
  i.no.intensity = FALSE,
  i.epidemic.start = NA,
  i.range.x = c(40, 20),
  i.range.x.53 = FALSE,
  i.range.y = NA,
  i.no.labels = FALSE,
  i.start.end.marks = TRUE,
  i.mem.info = TRUE
)
```

## Arguments

|                                     |                                                                                                                            |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| <code>i.current</code>              | Current season weekly rates.                                                                                               |
| <code>i.epidemic.thresholds</code>  | Pre and post epidemic thresholds.                                                                                          |
| <code>i.intensity.thresholds</code> | Intensity thresholds.                                                                                                      |
| <code>i.mean.length</code>          | Mean length of epidemic.                                                                                                   |
| <code>i.force.length</code>         | If you want to force the epidemic to be exactly as the mean length.                                                        |
| <code>i.output</code>               | Directory where graph is saved.                                                                                            |
| <code>i.graph.title</code>          | Title of the graph.                                                                                                        |
| <code>i.graph.subtitle</code>       | Subtitle of the graph.                                                                                                     |
| <code>i.graph.file</code>           | Graph to a file.                                                                                                           |
| <code>i.graph.file.name</code>      | Name of the graph.                                                                                                         |
| <code>i.week.report</code>          | Week to use in the report.                                                                                                 |
| <code>i.equal</code>                | If post epidemic and preepidemic thresholds must be equal (force post epidemic to be equal to the pre epidemic threshold). |
| <code>i.pos.epidemic</code>         | Print post epidemic threhsold.                                                                                             |
| <code>i.no.epidemic</code>          | Force no start of the epidemic, print only the epidemic threshold.                                                         |
| <code>i.no.intensity</code>         | Do not print intensity thresholds.                                                                                         |
| <code>i.epidemic.start</code>       | Week to force start of the epidemic.                                                                                       |
| <code>i.range.x</code>              | Range of weeks.                                                                                                            |
| <code>i.range.x.53</code>           | Is there a week 53 this season.                                                                                            |
| <code>i.range.y</code>              | Range of graph.                                                                                                            |
| <code>i.no.labels</code>            | Do not use labels.                                                                                                         |
| <code>i.start.end.marks</code>      | Do not place start and end marks of the epidemic.                                                                          |
| <code>i.mem.info</code>             | include information about the package in the graph.                                                                        |

## Details

Input data must be the current season and an object of class `mem`. The output graph contains the weekly rates series along with the epidemic and intensity thresholds located at the exact situation where the epidemic started. If there is no epidemic yet, only the epidemic threshold is placed.

Surveillance consist on:

1. Draw weekly values along with the pre-epidemic threshold.
2. When the weekly value rises above the threshold a marker of the start of the epidemic is placed and the intensity thresholds are drawn.

3. When the weekly value goes down the post-epidemic threshold, the marker of the end of the epidemic is placed and the post-epidemic threshold is added to the graph.

The Surveillance Week allows select the week to use in the surveillance, the values of the surveillance season are shown only up to this week, and the program will ignore values past this week.

The Force epidemic start allows to force the placement of the epidemic start marker at a given week instead of using the first week above the epidemic threshold.

## Value

memsurveillance writes a tiff graph of the surveillance of this season.

## Author(s)

Jose E. Lozano <lozalojo@gmail.com>

## References

Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? Int Congr Ser. 2004 Jun;1263:281-3.

Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. Influenza Other Respir Viruses. 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.

Vega T, Lozano JE, Meerhoff T, Snacken R, Beaute J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. Influenza Other Respir Viruses. 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.

Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983

## Examples

```
# Castilla y Leon Influenza Rates data
data(flucyl)
# Data of the last season
cur <- flucyl[8]
# The model
epi <- memmodel(flucyl[1:7])
# Epidemic thresholds
e.thr <- epi$epidemic.thresholds
# Intensity thresholds
i.thr <- epi$intensity.thresholds
# Set the working directory to wherever you want to store the graph file
setwd(".")
# The graph, default values
# uncomment to execute
# m1 <- memsurveillance(cur, e.thr, i.thr,
#   i.graph.file = TRUE,
#   i.graph.file.name = "graph 1"
# )
# No intensity levels
```

```

# m2 <- memsurveillance(cur, e.thr, i.thr,
#   i.graph.file = TRUE,
#   i.graph.file.name = "graph 2", i.no.intensity = TRUE
# )
# No start/end tickmarks
# m3 <- memsurveillance(cur, e.thr, i.thr,
#   i.graph.file = TRUE,
#   i.graph.file.name = "graph 3", i.start.end.marks = FALSE
# )
# Post-epidemic threshold
# m4 <- memsurveillance(cur, e.thr, i.thr,
#   i.graph.file = TRUE,
#   i.graph.file.name = "graph 4", i.pos.epidemic = TRUE
# )
# Report for week 2, instead of all data
# m5 <- memsurveillance(cur, e.thr, i.thr,
#   i.graph.file = TRUE,
#   i.graph.file.name = "graph 5", i.week.report = 2
# )

```

---

memsurveillance.animated

*Creates the animated graph of the surveillance of the current season*

---

## Description

Function `memsurveillance.animated` creates an animated surveillance graph for the current season.

## Usage

```

memsurveillance.animated(
  i.current,
  i.epidemic.thresholds = NA,
  i.intensity.thresholds = NA,
  i.output = ".",
  i.animated.graph.file = TRUE,
  i.animated.graph.file.name = "animated",
  i.fps = 2,
  i.loop = 0,
  i.remove = TRUE,
  ...
)

```

## Arguments

|                                         |                                                                         |
|-----------------------------------------|-------------------------------------------------------------------------|
| <code>i.current</code>                  | Current season weekly rates.                                            |
| <code>i.epidemic.thresholds</code>      | Pre and post epidemic thresholds.                                       |
| <code>i.intensity.thresholds</code>     | Intensity thresholds.                                                   |
| <code>i.output</code>                   | Directory where graph is saved.                                         |
| <code>i.animated.graph.file</code>      | If a animated gif should be produced, or just the intermediate graphics |
| <code>i.animated.graph.file.name</code> | Name of the animated graph.                                             |
| <code>i.fps</code>                      | Number of frames per second of the animated gif.                        |
| <code>i.loop</code>                     | Number of loops for the animated dif, 0 for Infinite.                   |
| <code>i.remove</code>                   | Remove partial graphs.                                                  |
| <code>...</code>                        | Additional parameters parsed to memsurveillance.                        |

## Details

Input data must be the current season and an object of class mem. The output graph contains the weekly rates series along with the epidemic and intensity threshols located at the exact situation where the epidemic started. If there is no epidemic yet, only the epidemic threshold is placed.

See [memsurveillance](#) for more details

## Value

`memsurveillance.animated` writes a gif graph of the surveillance of this season.

## Author(s)

Jose E. Lozano <lozalojo@gmail.com>

## References

- Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? Int Congr Ser. 2004 Jun;1263:281-3.
- Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. Influenza Other Respir Viruses. 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.
- Vega T, Lozano JE, Meerhoff T, Snacken R, Beaute J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. Influenza Other Respir Viruses. 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.
- Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983

## Examples

```
# Castilla y Leon Influenza Rates data
data(flucyl)
# Data of the last season
cur <- flucyl[8]
# The model
epi <- memmodel(flucyl[1:7])
# Epidemic thresholds
e.thr <- epi$epidemic.thresholds
# Intensity thresholds
i.thr <- epi$intensity.thresholds
# Set the working directory to wherever you want to store the graph file
setwd(".")
# uncomment to execute
# m1 <- memsurveillance.animated(cur,
#   i.animated.graph.file.name = "Animated",
#   i.epidemic.thresholds = e.thr, i.intensity.thresholds = i.thr, i.pos.epidemic = TRUE,
#   i.animated.graph.file = FALSE
# )
```

---

memtiming

*Influenza Epidemic Timing*


---

## Description

Function `memtiming` is used to find the optimal timing of an influenza epidemic in a set of weekly influenza surveillance rates. It provides the start and the end of the epidemic, also it returns a list of pre-epidemic and post-epidemic rates that can be used to calculate influenza baselines and thresholds.

## Usage

```
memtiming(
  i.data,
  i.n.values = 5,
  i.method = 2,
  i.param = 2.8,
  i.mem.info = TRUE
)
```

## Arguments

|                         |                                                                                                |
|-------------------------|------------------------------------------------------------------------------------------------|
| <code>i.data</code>     | a numeric object (or one that can be coerced to that class).                                   |
| <code>i.n.values</code> | a number, which indicates how many pre-epidemic values are taken from the pre-epidemic period. |
| <code>i.method</code>   | a number from 1 to 4, to select which optimization method to use.                              |

`i.param` an optional parameter used by the method.  
`i.mem.info` include information about the package in the graph.

## Details

The method to calculate the optimal timing of an epidemic is described as part of the *Moving Epidemics Method* (MEM), used to monitor influenza activity in a weekly surveillance system.

Input data is a vector of rates that represent a full influenza surveillance season. It can start and end at any week (typically at week 40th), and rates can be expressed as per 100,000 inhabitants (or per consultations, if population is not available) or any other scale.

The `i.n.values` parameter is used to get information from the pre-epidemic and post-epidemic period. The function will extract the highest pre/post values in order to use it later to calculate other influenza indicators, such as baseline activity or threshold for influenza epidemic.

Depending of the value `i.method`, the function will use a different method to calculate the optimum epidemic timing.

- 1 original method
- 2 fixed criterium method
- 3 slope method
- 4 second derivative method

All methods are based upon the MAP curve, as described in the MEM Method.

The *original method* uses the process shown in the original paper, which describes the MEM as it was created. The *fixed criterium method* is an update of the MEM that uses the slope of the MAP curve to find the optimum, which is the point where the slope is lower than a predefined value. The *slope method* also calculates the slope of the MAP curve, but the optimum is the one that matches the global/mean slope. The *second derivative method* calculates the second derivative and equals to zero to search an inflexion point in the original curve.

Two of the four methods require an additional parameter `i.param`: for the *fixed criterium method* is the predefined value to find the optimum, which typically is 2.5-3.0%, and for the *original method* it is needed the window parameter to smooth the map curve. A value of -1 indicates it should use [h.select](#) to select the window parameter. See [sm](#) for more information about this topic.

## Value

`memtiming` returns an object of class `epidemic`. An object of class `epidemic` is a list containing at least the following components:

**i.data** input data  
**data** data with missing rates filled with data from smoothing regression  
**map.curve** MAP curve  
**slope.curve** slope of the MAP curve  
**optimum.map** optimum  
**pre.epi** pre-epidemic highest rates  
**epi** epidemic highest rates

**post.epi** post-epidemic highest rates

**pre.epi.data** pre-epidemic rates

**epi.data** epidemic rates

**post.epi.data** post-epidemic rates

### Author(s)

Jose E. Lozano <lozalojo@gmail.com>

### References

Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? Int Congr Ser. 2004 Jun;1263:281-3.

Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. Influenza Other Respir Viruses. 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.

Vega T, Lozano JE, Meerhoff T, Snacken R, Beate J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. Influenza Other Respir Viruses. 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.

Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983

### Examples

```
# Castilla y Leon Influenza Rates data
data(flucyl)
# Finds the timing of the first season: 2001/2002
tim <- memtiming(flucyl[1])
print(tim)
summary(tim)
plot(tim)
```

---

memtrend

*Methods for influenza trend calculation*

---

### Description

Function memtrend is used to calculate the two parameters for defining the current influenza trend.

**Usage**

```
memtrend(
  i.flu,
  i.type = 1,
  i.level = 0.95,
  i.type.boot = "norm",
  i.iter.boot = 10000,
  i.use.t = FALSE
)
```

**Arguments**

|                          |                                                                                     |
|--------------------------|-------------------------------------------------------------------------------------|
| <code>i.flu</code>       | An object of class <code>mem</code> .                                               |
| <code>i.type</code>      | Type of confidence interval to calculate the trend thresholds.                      |
| <code>i.level</code>     | Level of confidence interval to calculate the trend thresholds.                     |
| <code>i.type.boot</code> | Type of bootstrap technique.                                                        |
| <code>i.iter.boot</code> | Number of bootstrap iterations.                                                     |
| <code>i.use.t</code>     | Whether to use t-values (t distribution) instead of z-values (normal distribution). |

**Details**

This method is based on the Moving Epidemics Method (MEM) used to monitor influenza activity in a weekly surveillance system.

Input data is a data frame containing rates that represent historical influenza surveillance data. It can start and end at any given week (typically at week 40th), and rates can be expressed as per 100,000 inhabitants (or per consultations, if population is not available) or any other scale.

The `i.seasons` parameter indicates how many seasons are used for calculating thresholds. A value of -1 indicates the program to use as many as possible. If there are less than this parameter, the program used all seasons available.

The `i.use.t` parameter allows to use the t-value (from a t distribution) instead of the z-value (from a normal distribution) for confidence intervals that rely on z/t-values. Useful when using less than 30 values to calculate confidence intervals.

There are three different states for trend, to determine the state, the current rate and the difference of the current and last weekly rate are needed:

- 2 Ascending - When the weekly rate is above the epidemic threshold and the difference of the current and last weekly rate is higher than Delta OR this is the first time the rate is above the epidemic threshold.
- 3 Descending - When the weekly rate is above the epidemic threshold and the difference of the current and last weekly rate is lower than Eta OR this is the first time the rate is below the epidemic threshold after having been above it.
- 1 Stable - Otherwise.

**Value**

memtrend returns a list with two objects, the first one is the parameter used in the calculations (param.seasons) and the second one (trend.thresholds) is a matrix 1x2 with the Ascending (Delta) and Descending parameters (Eta).

1. Delta - Ascending parameter.
2. Eta - Descending parameter.

**Author(s)**

Jose E. Lozano <lozalojo@gmail.com>

**References**

Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? Int Congr Ser. 2004 Jun;1263:281-3.

Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. Influenza Other Respir Viruses. 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.

Vega T, Lozano JE, Meerhoff T, Snacken R, Beate J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. Influenza Other Respir Viruses. 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.

Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983

**Examples**

```
# Castilla y Leon Influenza Rates data
data(flucyl)
# mem model
flucyl.mem <- memmodel(flucyl)
# Calculates trend thresholds
trend <- memtrend(flucyl.mem)
trend
```

---

optimum.by.inspection *Inspection calculation of the optimum*

---

**Description**

Function optimum.by.inspection perform an analysis of mem parameters to find the one that fits better a panel of experts inspection criterium.

To be written

**Usage**

```

optimum.by.inspection(
  i.data,
  i.param.values = seq(1, 5, 0.1),
  i.graph = FALSE,
  i.graph.file = FALSE,
  i.graph.file.name = "",
  i.graph.title = "",
  i.graph.subtitle = "",
  i.output = ".",
  i.labels.axis = c("Week", "Weekly rate"),
  i.labels.periods = c("Pre", "Epidemic", "Post"),
  i.labels.details = c("Sensitivity", "Specificity",
    "First row represents real and second row observed pos/neg values"),
  i.tolerance = 0.25
)

```

**Arguments**

|                                |                                                                             |
|--------------------------------|-----------------------------------------------------------------------------|
| <code>i.data</code>            | Data frame of input data.                                                   |
| <code>i.param.values</code>    | values to use in the <code>i.param</code> value of <code>memtiming</code> . |
| <code>i.graph</code>           | create a graph with the outputs (T/FALSE).                                  |
| <code>i.graph.file</code>      | write the graph to a file.                                                  |
| <code>i.graph.file.name</code> | name of the output file.                                                    |
| <code>i.graph.title</code>     | title of the graph.                                                         |
| <code>i.graph.subtitle</code>  | subtitle of the graph.                                                      |
| <code>i.output</code>          | output directory.                                                           |
| <code>i.labels.axis</code>     | different labels used by output graphs                                      |
| <code>i.labels.periods</code>  | different labels used by output graphs                                      |
| <code>i.labels.details</code>  | different labels used by output graphs                                      |
| <code>i.tolerance</code>       | tolerance (in inches) when identifying points in the graph                  |

**Value**

`optimum.by.inspection` returns a list. An object of class `mem` is a list containing at least the following components:

|                              |                                     |
|------------------------------|-------------------------------------|
| <code>optimum</code>         | optimum value.                      |
| <code>optimum.data</code>    | Data related to the optimum value.  |
| <code>summary.data</code>    | Data for all values tested.         |
| <code>inspection.data</code> | Detailed results of each iteration. |

**Author(s)**

Jose E. Lozano <lozalojo@gmail.com>

**References**

Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? Int Congr Ser. 2004 Jun;1263:281-3.

Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. Influenza Other Respir Viruses. 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.

Vega T, Lozano JE, Meerhoff T, Snacken R, Beaute J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. Influenza Other Respir Viruses. 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.

Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983

**Examples**

```
# Castilla y Leon Influenza Rates data
data(flucyl)
# Inspection. It runs interactively (uncomment to run)
# if (interactive()) {
#   opt.ins <- optimum.by.inspection(flucyl, i.param.values = seq(2.0, 3.0, 0.1))
#   opt.ins$optimum
# }
```

---

processPlots

*Full process plots for mem*

---

**Description**

Function processPlots creates graphs of all mem process

**Usage**

```
processPlots(i.flu, i.output = ".", i.prefix = "")
```

**Arguments**

|          |                                    |
|----------|------------------------------------|
| i.flu    | Object of class flu.               |
| i.output | Output directory.                  |
| i.prefix | Prefix for all files to be output. |

## Details

Create plots related to the process of calculating MEM indicators to an output directory, showing the MAP curves and the slope of the MAP curve and how the timing is calculated.

## Value

processPlots prints a set of graphs.

## Author(s)

Jose E. Lozano <lozalojo@gmail.com>

## References

Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? Int Congr Ser. 2004 Jun;1263:281-3.

Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. Influenza Other Respir Viruses. 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.

Vega T, Lozano JE, Meerhoff T, Snacken R, Beaute J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. Influenza Other Respir Viruses. 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.

Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983

## Examples

```
# Castilla y Leon Influenza Rates data
data(flucyl)
# Graphs
epi <- memmodel(flucyl)
# uncomment to execute
# processPlots(epi)
```

---

roc.analysis

*Analysis of different indicators to find the optimum value of the window parameter*

---

## Description

Function roc.analysis perform a ROC analysis

**Usage**

```
roc.analysis(
  i.data,
  i.param.values = seq(1, 5, 0.1),
  i.min.seasons = 6,
  i.graph = FALSE,
  i.graph.file = FALSE,
  i.graph.file.name = "",
  i.graph.title = "",
  i.graph.subtitle = "",
  i.output = ".",
  i.mem.info = TRUE,
  i.min.sensitivity = NA,
  i.min.specificity = NA,
  ...
)
```

**Arguments**

|                                |                                                               |
|--------------------------------|---------------------------------------------------------------|
| <code>i.data</code>            | Data frame of input data.                                     |
| <code>i.param.values</code>    | range of <code>i.param</code> values to test.                 |
| <code>i.min.seasons</code>     | minimum number of seasons to perform the analysis, default=6. |
| <code>i.graph</code>           | create a graph with the outputs (TRUE/FALSE).                 |
| <code>i.graph.file</code>      | write the graph to a file.                                    |
| <code>i.graph.file.name</code> | name of the output file.                                      |
| <code>i.graph.title</code>     | title of the graph.                                           |
| <code>i.graph.subtitle</code>  | subtitle of the graph.                                        |
| <code>i.output</code>          | output directory.                                             |
| <code>i.mem.info</code>        | include information about the package in the graph.           |
| <code>i.min.sensitivity</code> | set a minimum sensitivity when finding the optimum            |
| <code>i.min.specificity</code> | set a minimum specificity when finding the optimum            |
| <code>...</code>               | other paramaters to be used by memgoodness function.          |

**Details**

Optimize is an iterative process that calculates goodness indicators using different window parameters for the fixed criterium and compares all estimators in order to find the optimum window parameter.

The output shows the different window parameters and their respective indicators to decide which one is better for your data.

**Value**

roc.analysis returns a list. An object of class mem is a list containing at least the following components:

**optimum** optimum value.

**results** Detailed results of each iteration.

**Author(s)**

Jose E. Lozano <lozalojo@gmail.com>

**References**

Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? Int Congr Ser. 2004 Jun;1263:281-3.

Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. Influenza Other Respir Viruses. 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.

Vega T, Lozano JE, Meerhoff T, Snacken R, Beaute J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. Influenza Other Respir Viruses. 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.

Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983

**Examples**

```
# Castilla y Leon Influenza Rates data
data(flucyl)
# ROC analysis
epi.roc <- roc.analysis(flucyl,
  i.param.values = seq(2.6, 2.8, 0.1),
  i.detection.values = seq(2.6, 2.8, 0.1)
)
epi.roc$results
```

---

transformdata

*Data transformation*


---

**Description**

Function transformdata transforms data from year,week,rate to week,rate1,...,rateN suitable to use with mem.

**Usage**

```
transformdata(
  i.data,
  i.range.x = NA,
  i.name = "rates",
  i.max.na.per = 100,
  i.function = NULL
)
```

**Arguments**

|                           |                                                                                                                                        |
|---------------------------|----------------------------------------------------------------------------------------------------------------------------------------|
| <code>i.data</code>       | Data frame of input data.                                                                                                              |
| <code>i.range.x</code>    | First and last surveillance week.                                                                                                      |
| <code>i.name</code>       | Name of the column to transform.                                                                                                       |
| <code>i.max.na.per</code> | maximum percentage of na's in a season allowable, otherwise, the season is removed                                                     |
| <code>i.function</code>   | function used to aggregate data when duplicate values are found for the same season and week, defaults to NULL (no aggregate function) |

**Details**

Transform data from format year, week, rate to a format suitable to be used with [mem](#), that is, one column with each season in the dataset and the week names as rownames of the data.frame.

**Value**

`transformdata` returns a data.frame where each column has a different season and rownames are the name of the epidemiological week.

**Author(s)**

Jose E. Lozano <lozalojo@gmail.com>

**References**

Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? *Int Congr Ser.* 2004 Jun;1263:281-3.

Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. *Influenza Other Respir Viruses.* 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.

Vega T, Lozano JE, Meerhoff T, Snacken R, Beaute J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. *Influenza Other Respir Viruses.* 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.

Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983

**Examples**

```
# Castilla y Leon Influenza Rates data
data(flucylraw)
# Transform data
newdata <- transformdata(flucylraw, i.range.x = c(40, 20))$data
epi <- memmodel(newdata)
print(epi)
summary(epi)
plot(epi)
```

---

|                    |                            |
|--------------------|----------------------------|
| transformdata.back | <i>Data transformation</i> |
|--------------------|----------------------------|

---

**Description**

Function transformdata.back transforms data from week,rate1,...,rateN to year,week,rate format.

**Usage**

```
transformdata.back(
  i.data,
  i.name = "rates",
  i.cutoff.original = NA,
  i.range.x.final = NA,
  i.fun = mean
)
```

**Arguments**

|                   |                                                        |
|-------------------|--------------------------------------------------------|
| i.data            | Data frame of input data.                              |
| i.name            | Name of the column that contains the values.           |
| i.cutoff.original | Cutoff point between seasons when they have two years  |
| i.range.x.final   | Range of the surveillance period in the output dataset |
| i.fun             | sumarize function                                      |

**Details**

Transforms data from the season in each column format (the one that uses [mem](#)) to the format year, week, rate in a 3 columns data.frame.

Allows to set the cutoff point to separate between two seasons when one season has two different years.

**Value**

transformdata.back returns a data.frame with three columns, year, week and rate.

**Author(s)**

Jose E. Lozano <lozalojo@gmail.com>

**References**

Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? Int Congr Ser. 2004 Jun;1263:281-3.

Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. Influenza Other Respir Viruses. 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.

Vega T, Lozano JE, Meerhoff T, Snacken R, Beaute J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. Influenza Other Respir Viruses. 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.

Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983

**Examples**

```
# Castilla y Leon Influenza Rates data
data(flucyl)
# Transform data
newdata <- transformdata.back(flucyl)$data
```

---

|                 |                                         |
|-----------------|-----------------------------------------|
| transformseries | <i>Transformation of series of data</i> |
|-----------------|-----------------------------------------|

---

**Description**

Function transformseries transforms whole datasets.

**Usage**

```
transformseries(i.data, i.transformation = 1, ...)
```

**Arguments**

|                  |                                                            |
|------------------|------------------------------------------------------------|
| i.data           | Historical data series.                                    |
| i.transformation | Transformation to apply to the dataset.                    |
| ...              | other paramaters to be used by functions to fit two waves. |

**Details**

Input data must be a data.frame with each column a surveillance season and each row a week.

Transformation options:

- 1 No transformation
- 2 Odd
- 3 Fill missing data
- 4 Smoothing regression
- 5 Two waves (observed)
- 6 Two waves (expected)
- 7 Loess
- 8 Spline
- 9 Moving Average

Fill missings substitute missing values with predicted values from a loess regression fit. If does not impute leading or trailing missings, only missings in the middle of the season.

Odd calculates the odd:  $p$  divided by  $1-p$ .

Loess substitute the dataset with predicted values from a loess regression fit.

Spline fits a cubic smoothing spline to the supplied data.

Two waves (observed) is used when there are two waves per season. It divides the original dataset in two using a mixture of two normal distributions. The expected option uses the same procedure but also substitutes all data with predicted values of the mixture fit.

Moving Average has an additional parameter `i.number` for the number of observation to calculate the average.

**Value**

`transformseries` The transformed dataset.

**Author(s)**

Jose E. Lozano <lozalojo@gmail.com>

**References**

Vega T, Lozano JE, Ortiz de Lejarazu R, Gutierrez Perez M. Modelling influenza epidemic - can we detect the beginning and predict the intensity and duration? *Int Congr Ser.* 2004 Jun;1263:281-3.

Vega T, Lozano JE, Meerhoff T, Snacken R, Mott J, Ortiz de Lejarazu R, et al. Influenza surveillance in Europe: establishing epidemic thresholds by the moving epidemic method. *Influenza Other Respir Viruses.* 2013 Jul;7(4):546-58. DOI:10.1111/j.1750-2659.2012.00422.x.

Vega T, Lozano JE, Meerhoff T, Snacken R, Beate J, Jorgensen P, et al. Influenza surveillance in Europe: comparing intensity levels calculated using the moving epidemic method. *Influenza Other Respir Viruses.* 2015 Sep;9(5):234-46. DOI:10.1111/irv.12330.

Lozano JE. lozalojo/mem: Second release of the MEM R library. Zenodo [Internet]. [cited 2017 Feb 1]; Available from: <https://zenodo.org/record/165983>. DOI:10.5281/zenodo.165983

**Examples**

```
# Castilla y Leon Influenza Rates data
data(flucyl)
# Data of the last season
transformseries(flucyl, 2)
```

# Index

## \* datasets

flucyl, [4](#)

flucylraw, [5](#)

## \* influenza

full.series.graph, [6](#)

memevolution, [8](#)

memgoodness, [10](#)

memintensity, [14](#)

memmodel, [15](#)

memstability, [20](#)

memsurveillance, [21](#)

memsurveillance.animated, [24](#)

mentiming, [26](#)

mentrend, [28](#)

optimum.by.inspection, [30](#)

processPlots, [32](#)

roc.analysis, [33](#)

transformdata, [35](#)

transformdata.back, [37](#)

transformseries, [38](#)

epimem, [2](#)

epitiming, [3](#)

flucyl, [4](#)

flucylraw, [5](#)

full.series.graph, [6](#)

h.select, [27](#)

mem, [36](#), [37](#)

memevolution, [8](#)

memgoodness, [10](#)

memintensity, [14](#)

memmodel, [3](#), [9](#), [15](#)

memstability, [20](#)

memsurveillance, [21](#), [25](#)

memsurveillance.animated, [24](#)

mentiming, [3](#), [17](#), [26](#)

mentrend, [28](#)

optimum.by.inspection, [30](#)

processPlots, [32](#)

roc.analysis, [33](#)

sm, [27](#)

summary.epidemic, plot.epidemic, print.epidemic  
(mentiming), [26](#)

summary.flu, plot.flu, print.flu  
(memmodel), [15](#)

transformdata, [35](#)

transformdata.back, [37](#)

transformseries, [38](#)