

# Package ‘R0’

July 21, 2025

**Type** Package

**Title** Estimation of R0 and Real-Time Reproduction Number from Epidemics

**Version** 1.3-1

**Date** 2023-09-26

**Author** Pierre-Yves Boelle, Thomas Obadia

**URL** <https://github.com/tobadia/R0>

**Maintainer** Thomas Obadia <thomas.obadia@pasteur.fr>

**Depends** R (>= 2.13.0)

**Imports** MASS

**Description** Estimation of reproduction numbers for disease outbreak, based on incidence data. The R0 package implements several documented methods. It is therefore possible to compare estimations according to the methods used. Depending on the methods requested by user, basic reproduction number (commonly denoted as R0) or real-time reproduction number (referred to as R(t)) is computed, along with a 95% Confidence Interval. Plotting outputs will give different graphs depending on the methods requested : basic reproductive number estimations will only show the epidemic curve (collected data) and an adjusted model, whereas real-time methods will also show the R(t) variations throughout the outbreak time period. Sensitivity analysis tools are also provided, and allow for investigating effects of varying Generation Time distribution or time window on estimates.

**License** GPL (>= 2)

**LazyLoad** yes

**RoxygenNote** 7.2.3

**Encoding** UTF-8

**NeedsCompilation** no

**Repository** CRAN

**Date/Publication** 2023-09-26 09:50:02 UTC

## Contents

|                                |    |
|--------------------------------|----|
| est.GT . . . . .               | 2  |
| estimate.R . . . . .           | 3  |
| generation.time . . . . .      | 6  |
| Germany.1918 . . . . .         | 8  |
| GT.chld.hsld . . . . .         | 8  |
| H1N1.serial.interval . . . . . | 9  |
| impute.incid . . . . .         | 10 |
| inspect.data . . . . .         | 10 |
| integrity.checks . . . . .     | 12 |
| sa.GT . . . . .                | 13 |
| sa.time . . . . .              | 15 |
| sensitivity.analysis . . . . . | 16 |
| sim.epid . . . . .             | 19 |
| sim.epid.indiv . . . . .       | 20 |
| smooth.Rt . . . . .            | 21 |

|              |           |
|--------------|-----------|
| <b>Index</b> | <b>24</b> |
|--------------|-----------|

---

|        |  |
|--------|--|
| est.GT | <i>Estimate generation time distribution</i> |
|--------|--|

---

### Description

Find the best-fitting generation time distribution from a series of serial interval.

### Usage

```
est.GT(
  infector.onset.dates = NULL,
  infectee.onset.dates = NULL,
  serial.interval = NULL,
  request.plot = FALSE,
  ...
)
```

### Arguments

|                      |  |
|----------------------|--|
| infector.onset.dates | Vector of dates for infector symptoms onset.   |
| infectee.onset.dates | Vector of dates for infectee symptoms onset.   |
| serial.interval      | Vector of reported serial interval.  |
| request.plot         | Boolean. Should data adjustment be displayed at the end?   |
| ...                  | Parameters passed to other functions (useful for hidden parameters of <code>generation.time()</code> ) |

## Details

Generation Time (GT) distribution can be estimated by two inputs methods. User can either provide two vectors of dates or a unique vector of reported serial intervals. If two vectors are provided, both `*.onset.dates` vectors should have the same length. The  $i$ -th element is the onset date for individual  $i$ . This implies that infector  $k$  (symptoms on day `infector.onset.dates[k]`) infected infectee  $k$  (symptoms on day `infectee.onset.dates[k]`). If only `serial.interval` is provided, each value is assumed to be the time elapsed between each pair of infector and infectee (i.e. the difference between the two `*.onset.dates` vectors).

When `request.plot` is set to TRUE, a graphical output provides standardized histogram of observed data along with the best-fitting adjusted model.

## Value

A object of class `R0.GT` that complies with `generation.time()` distribution requirements of the R0 package.

## Author(s)

Pierre-Yves Boelle, Thomas Obadia

## Examples

```
#Loading package
library(R0)

# Data taken from traced cases of H1N1 viruses.
data(H1N1.serial.interval)
est.GT(serial.interval=H1N1.serial.interval)

## Best fitting GT distribution is a gamma distribution with mean = 3.039437 and sd = 1.676551 .
## Discretized Generation Time distribution
## mean: 3.070303 , sd: 1.676531
## [1] 0.0000000000 0.1621208802 0.2704857362 0.2358751176 0.1561845680 0.0888997193 0.0459909903
## 0.0222778094 0.0102848887 0.0045773285 0.0019791984 0.0008360608 0.0003464431 0.0001412594

# The same result can be achieved with two vectors of dates of onset.
# Here we use the same data, but trick the function into thinking onset dates are all "0".
data(H1N1.serial.interval)
est.GT(infector.onset.dates=rep(0,length(H1N1.serial.interval)),
      infectee.onset.dates=H1N1.serial.interval)
```

---

estimate.R

*Estimate reproduction number ( $R_0$  or  $R_t$ ) for one incidence dataset using available methods*

---

## Description

Estimate  $R_0$  or  $R(t)$  for an incidence dataset using the methods implemented in the R0 package.

**Usage**

```
estimate.R(
  epid = NULL,
  GT = NULL,
  t = NULL,
  begin = NULL,
  end = NULL,
  date.first.obs = NULL,
  time.step = 1,
  AR = NULL,
  pop.size = NULL,
  S0 = 1,
  methods = NULL,
  checked = TRUE,
  ...
)
```

**Arguments**

|                |   |
|----------------|---|
| epid           | Object containing epidemic curve data.  |
| GT             | Generation time distribution from <a href="#">generation.time()</a> .   |
| t              | Vector of dates at which incidence was observed.  |
| begin          | Begin date for estimation. Can be an integer or a date (YYYY-mm-dd or YYYY/mm/dd).  |
| end            | End date for estimation. Can be an integer or a date (YYYY-mm-dd or YYYY/mm/dd).  |
| date.first.obs | Optional date of first observation, if t not specified.   |
| time.step      | Optional. If date of first observation is specified, number of day between each incidence observation.                    |
| AR             | Attack rate as a percentage from total population.  |
| pop.size       | Population size in which the incident cases were observed. See more details in <a href="#">est.R0.AR()</a> documentation. |
| S0             | Initial proportion of the population considered susceptible.  |
| methods        | Vector of methods to be used for R/R0/Rt estimation. Must be provided as <code>c("method 1", "method 2", ...)</code> .    |
| checked        | Internal flag used to check whether integrity checks were ran or not.   |
| ...            | Parameters passed to inner functions.   |

**Details**

Currently, supported methods are Exponential Growth (EG), Maximum Likelihood (ML), Attack Rate (AR), Time-Dependant (TD), and Sequential Bayesian (SB). The corresponding references from the literature are available below.

This function acts as a front-end and will prepare relevant inputs to pass them to internal estimation routines. In particular, all inputs will undergo validation through [integrity.checks\(\)](#) and the checked flag (defaulting as TRUE here) will be passed to internal estimation routines. Any warning raised by [integrity.checks\(\)](#) should warrant careful thinking and investigation.

**Value**

A list with components:

|           |  |
|-----------|--|
| estimates | List containing all results from called methods. |
| epid      | Epidemic curve.                                  |
| GT        | Generation Time distribution function.           |
| t         | Date vector.                                     |
| begin     | Begin date for estimation.                       |
| end       | End date for estimation.                         |

**Author(s)**

Pierre-Yves Boelle, Thomas Obadia

**References**

[est.R0.AR\(\)](#) : Dietz, K. "The Estimation of the Basic Reproduction Number for Infectious Diseases." *Statistical Methods in Medical Research* 2, no. 1 (March 1, 1993): 23-41.

[est.R0.EG\(\)](#) : Wallinga, J., and M. Lipsitch. "How Generation Intervals Shape the Relationship Between Growth Rates and Reproductive Numbers." *Proceedings of the Royal Society B: Biological Sciences* 274, no. 1609 (2007): 599.

[est.R0.ML\(\)](#) : White, L.F., J. Wallinga, L. Finelli, C. Reed, S. Riley, M. Lipsitch, and M. Pagano. "Estimation of the Reproductive Number and the Serial Interval in Early Phase of the 2009 Influenza A/H1N1 Pandemic in the USA." *Influenza and Other Respiratory Viruses* 3, no. 6 (2009): 267-276.

[est.R0.SB\(\)](#) : Bettencourt, L.M.A., and R.M. Ribeiro. "Real Time Bayesian Estimation of the Epidemic Potential of Emerging Infectious Diseases." *PLoS One* 3, no. 5 (2008): e2185.

[est.R0.TD\(\)](#) : Wallinga, J., and P. Teunis. "Different Epidemic Curves for Severe Acute Respiratory Syndrome Reveal Similar Impacts of Control Measures." *American Journal of Epidemiology* 160, no. 6 (2004): 509; Cauchemez S., and Valleron AJ. "Estimating in Real Time the Efficacy of Measures to Control Emerging Communicable Diseases" *American Journal of Epidemiology* 164, no. 6 (2006): 591.

**Examples**

```
#Loading package
library(R0)

## Outbreak during 1918 influenza pandemic in Germany)
data(Germany.1918)
mGT <- generation.time("gamma", c(3, 1.5))
estR0 <- estimate.R(Germany.1918, mGT, begin=1, end=27, methods=c("EG", "ML", "TD", "AR", "SB"),
  pop.size=100000, nsim=100)

attributes(estR0)
## $names
## [1] "epid"      "GT"        "begin"     "end"       "estimates"
##
## $class
## [1] "R0.sR"
```

```

## Estimates results are stored in the $estimates object
estR0
## Reproduction number estimate using Exponential Growth method.
## R : 1.525895[ 1.494984 , 1.557779 ]
##
## Reproduction number estimate using Maximum Likelihood method.
## R : 1.383996[ 1.309545 , 1.461203 ]
##
## Reproduction number estimate using Attack Rate method.
## R : 1.047392[ 1.046394 , 1.048393 ]
##
## Reproduction number estimate using Time-Dependent method.
## 2.322239 2.272013 1.998474 1.843703 2.019297 1.867488 1.644993 1.553265 1.553317 1.601317 ...
##
## Reproduction number estimate using Sequential Bayesian method.
## 0 0 2.22 0.66 1.2 1.84 1.43 1.63 1.34 1.52 ...

## If no date vector nor date of first observation are provided, results are the same
## except time values in $t are replaced by index

```

---

generation.time

*Generation Time distribution*

---

## Description

Create an object of class GT representing a discretized Generation Time distribution, that can be subsequently passed to estimation routines.

## Usage

```

generation.time(
  type = c("empirical", "gamma", "weibull", "lognormal"),
  val = NULL,
  truncate = NULL,
  step = 1,
  first.half = TRUE,
  p0 = TRUE
)

```

## Arguments

|          |  |
|----------|--|
| type     | Type of distribution (can be any of "empirical", "gamma", "weibull", or "lognormal") |
| val      | Vector of values used for the empirical distribution, or c(mean, sd) if parametric.  |
| truncate | Maximum extent of the GT distribution.   |

|            |  |
|------------|--|
| step       | Time step used in discretization.  |
| first.half | Boolean. When set to TRUE (default), the first probability is computed on a half period. |
| p0         | Boolean. When set to TRUE the probability on day 0 is forced to 0.                       |

### Details

How the GT is discretized may have some impact on the shape of the distribution. For example, the distribution may be discretized in intervals of 1 time step starting at time 0, i.e. [0,1), [1,2), and so on. Or it may be discretized as [0,0.5), [0.5, 1.5), ... (the default).

If the GT is discretized from a given continuous distribution, the expected duration of the Generation Time will be less than the nominal, it will be in better agreement using the second discretization (default behavior).

If p0 is set to TRUE (default), the generation time distribution is set to 0 for day 0, meaning that the infectees generated by an infected individual will not become incident on the same day.

If no truncation is provided, the distribution will be truncated at 99.99% probability.

### Value

A list with components:

|      |   |
|------|---|
| GT   | The probabilities for each time unit, starting at time 0. |
| time | The time at which probabilities are calculated.           |
| mean | The mean of the discretized GT.                           |
| sd   | The standard deviation of the discretized GT.             |

### Author(s)

Pierre-Yves Boelle, Thomas Obadia

### Examples

```
#Loading package
library(R0)

# GT for children at house(from Cauchemez PNAS 2011)

GT.chld.hsld1 <- generation.time("empirical", c(0,0.25,0.2,0.15,0.1,0.09,0.05,0.01))
plot(GT.chld.hsld1, col="green")
GT.chld.hsld1
# Discretized Generation Time distribution
# mean: 2.729412 , sd: 1.611636
# [1] 0.00000000 0.29411765 0.23529412 0.17647059 0.11764706 0.10588235 0.05882353
# [8] 0.01176471

GT.chld.hsld2 <- generation.time("gamma", c(2.45, 1.38))
GT.chld.hsld2
# Discretized Generation Time distribution
# mean: 2.504038 , sd: 1.372760
```

```
# [1] 0.0000000000 0.2553188589 0.3247178420 0.2199060781 0.1144367560
# [6] 0.0515687896 0.0212246257 0.0082077973 0.0030329325 0.0010825594
#[11] 0.0003760069 0.0001277537
```

```
# GT for school & community
GTs1 <- generation.time("empirical", c(0,0.95,0.05))
plot(GTs1, col='blue')
```

```
plot(GT.chld.hsld1, ylim=c(0,0.5), col="red")
par(new=TRUE)
plot(GT.chld.hsld2, xlim=c(0,7), ylim=c(0,0.5), col="black")
```

---

Germany.1918

*Germany 1918 dataset*

---

### Description

This dataset is used as an example in the R0 package to estimate reproduction ratios using the available methods. It is the empirical distribution of Spanish flu in Prussia, Germany, from the 1918-19 influenza epidemic.

### Usage

```
data(Germany.1918)
```

### Format

A vector of numeric values named with date of record.

### References

Peiper O: Die Grippe-Epidemie in Preussen im Jahre 1918/19. Veroeffentlichungen aus dem Gebiete der Medizinalverwaltung 1920, 10: 417-479 (in German).

Nishiura H. Time variations in the transmissibility of pandemic influenza in Prussia, Germany, from 1918-19. In: Theoretical Biology and Medical Modelling

---

GT.chld.hsld

*2009 A/H1N1 observed generation time distribution*

---

### Description

Observed generation time distribution for children in household for the 2009 A/H1N1 influenza pandemic.



**Usage**

```
data(GT.chld.hsld)
```

**Format**

A vector of unnamed numeric values.

**References**

S. Cauchemez, A. Bhattarai, T.L. Marchbanks, R.P. Fagan, S. Ostroff, N.M. Ferguson, et al., Role of Social Networks in Shaping Disease Transmission During a Community Outbreak of 2009 H1N1 Pandemic Influenza, *Pnas*. 108 (2011) 2825-2830.

---

*H1N1.serial.interval*    *H1N1 serial interval*

---

**Description**

Serial interval (in its core definition, i.e. the time delay between symptom onset between pairs of infectors and infectees), taken from traced cases of H1N1 viruses.

Vector of values that represents the time lag between symptoms onset for pairs of infector/infectee, for a dataset of complete traced cases. Each value accounts for a pair of infector/infectee. This serial interval is often substituted for the generation time distribution, as it is easier to observe.

**Usage**

```
data(H1N1.serial.interval)
```

**Format**

A vector of unnamed numeric values.

**References**

S. Cauchemez, A. Bhattarai, T.L. Marchbanks, R.P. Fagan, S. Ostroff, N.M. Ferguson, et al., Role of Social Networks in Shaping Disease Transmission During a Community Outbreak of 2009 H1N1 Pandemic Influenza, *Pnas*. 108 (2011) 2825-2830.

---

|              |  |
|--------------|--|
| impute.incid | <i>Impute censored cases to rebuild longer epidemic vector</i> |
|--------------|--|

---

### Description

When first records of incidence are unavailable, tries to impute censored cases to rebuild a longer epidemic vector.

### Usage

```
impute.incid(CD.optim.vect, CD.epid, CD.R0, CD.GT)
```

### Arguments

|               |   |
|---------------|---|
| CD.optim.vect | Vector of two elements (c(multiplicative factor, log(highest imputed data))) to be optimized. |
| CD.epid       | Original epidemic vector, output of <a href="#">check.incid()</a> .                           |
| CD.R0         | Assumed R0 value for the original epidemic vector.  |
| CD.GT         | Generation time distribution to be used for computations.                                     |

### Details

This function is not intended for stand-alone use. It optimizes the values of vect, based upon minimization of deviation between actual epidemics data and observed generation time. The optimized function is [censored.deviation\(\)](#), which returns the deviation used for minimization. Stand-alone use can be conducted, however this assumes data are all of the correct format.

### Value

A vector with both imputed incidence and source available data.

### Author(s)

Pierre-Yves Boelle, Thomas Obadia

---

|              |   |
|--------------|---|
| inspect.data | <i>Audit input data for common issues</i> |
|--------------|---|

---

### Description

Inspect input data and look for common mistakes. The function does not return any object but yields warnings wupon detecting possible inconsistencies, along with suggestions as to how to clean inputs before running estimation routines.

**Usage**

```
inspect.data(incid, GT = NULL, t = NULL)
```

**Arguments**

|       |  |
|-------|--|
| incid | An object (vector, data.frame, list) storing incidence.            |
| GT    | Generation time distribution from <code>generation.time()</code> . |
| t     | Vector of dates at which incidence was observed (optional).        |

**Details**

`inspect.data()` looks for common issues that could affect estimation routines. Such issues include too low incidence counts, leading/trailing zeros, non-integer values...

Before any checks are conducted, the data are passed to `check.incid()` to try and guess the format of the data.

A not-so-uncommon issue is to provide non-integer counts for incidence, for example when working with aggregated data that represent averaged number of cases across different communities. This however does not agree well with parametric likelihood that assume exponential growth over the early stage of an epidemic or Poisson distribution of cases, where non-integer values will cause calculations to fail.

Missing values may cause issues if not handled properly. By default, `check.incid()` will recast missing values to zero. Leading and trailing NA's should be omitted entirely from the input. Gaps found between available data may also cause issues if they span over a period that's longer than the total generation time. A warning is raised to inform on these possible issues.

Likewise, leading and trailing zeros would cause similar issues. Begin will default to the first value and end to the peak one. Just in case, these will be inspected here too. Sequence of 0s exceeding the length of the generation time will also yield a warning.

Scarce data may also cause errors when optimizing likelihood functions. A time-series of incidence spanning for a duration shorter than that of the generation time distribution is likely to correspond to an index case that hasn't yet infected all its offsprings. This would bias estimates downwards and should be taken into account when interpreting results.

**Value**

No object is returned. Instead, warnings are thrown upon detecting inconsistencies.

**Author(s)**

Pierre-Yves Boelle, Thomas Obadia

---

integrity.checks      *Integrity checks for input parameters*

---

## Description

Before any requested estimation routine is ran, `integrity.checks()` is called to ensure the data passed as arguments meet the proper format and can be properly interpreted by subsequent functions.

## Usage

```
integrity.checks(
  epid,
  GT,
  t,
  begin,
  end,
  date.first.obs,
  time.step,
  AR,
  S0,
  methods
)
```

## Arguments

|                             |  |
|-----------------------------|--|
| <code>epid</code>           | Epidemic dataset, expecting incidence counts in a variety of possible formats (see <code>check.incid()</code> ).       |
| <code>GT</code>             | Generation time distribution from <code>generation.time()</code> .   |
| <code>t</code>              | Vector of dates at which incidence was observed.   |
| <code>begin</code>          | Begin date for estimation. Can be an integer or a date (YYYY-mm-dd or YYYY/mm/dd).                                     |
| <code>end</code>            | End date for estimation. Can be an integer or a date (YYYY-mm-dd or YYYY/mm/dd).                                       |
| <code>date.first.obs</code> | Optional date of first observation, if <code>t</code> not specified.   |
| <code>time.step</code>      | Optional. If date of first observation is specified, number of day between each incidence observation.                 |
| <code>AR</code>             | Attack rate as a percentage from total population.   |
| <code>S0</code>             | Initial proportion of the population considered susceptible.   |
| <code>methods</code>        | Vector of methods to be used for R/R0/Rt estimation. Must be provided as <code>c("method 1", "method 2", ...)</code> . |

## Details

For internal use. Called by all implemented estimation methods. All integrity/class checks are handled by this core function. GT must be an object of class `R0.GT`, and epidemic curve along with time values are handled here. If you plan on calling manually any other estimation function, make sure data are provided with correct format.

The epidemic curve `epid` may be provided as a vector. In that case, a vector `t` may be provided with the dates of observation. If `t` is not numeric, an attempt is made to convert to dates with `as.Date()`. If `t` is not provided, dates are obtained from the names of `incid`, and if not available, index values are used. Finally, one can provide an epidemic curve object generated by the `epitools` package (see `check.incid()` for more details).

A quick note on `t`, `begin` and `end` : when a date vector is provided (`t`), it will be used instead of index values to establish a date-related incidence. If no date vector is provided, then `begin` and `end` can still be forced to *numeric* values. It then links to the corresponding index values for incidence data. If a date vector is provided, `begin` and `end` can either be *numeric* values or *dates*. If numeric, they will link to the corresponding index values for incidence, and be afterward interpreted as the associated date. If date, they will be directly associated to incidence data.

Basically, if specified, `begin` and `end` must always have the same class.

## Value

A list with two components, `begin` and `end`.

## Author(s)

Pierre-Yves Boelle, Thomas Obadia

---

sa.GT

*Sensitivity of  $R_0$  to varying generation time distributions*

---

## Description

Sensitivity analysis to estimate the variation of reproduction numbers according to the distribution of generation time.

## Usage

```
sa.GT(
  incid,
  GT.type,
  GT.mean.range,
  GT.sd.range,
  begin = NULL,
  end = NULL,
  est.method,
  t = NULL,
  date.first.obs = NULL,
```

```

    time.step = 1,
    ...
)

```

### Arguments

|                             |   |
|-----------------------------|---|
| <code>incid</code>          | A vector of incident cases.   |
| <code>GT.type</code>        | Type of distribution for GT (see GT.R for details).   |
| <code>GT.mean.range</code>  | Range of mean values used for all GT distributions throughout the simulation. Must be provided as a vector. |
| <code>GT.sd.range</code>    | Range of standard deviation values used for GT distributions. Must be provided as a vector.                 |
| <code>begin</code>          | Vector of begin dates for the estimation of epidemic.   |
| <code>end</code>            | Vector of end dates for estimation of the epidemic.   |
| <code>est.method</code>     | Estimation method used for sensitivity analysis.  |
| <code>t</code>              | Dates vector to be passed to estimation function.   |
| <code>date.first.obs</code> | Optional date of first observation, if <code>t</code> not specified.  |
| <code>time.step</code>      | Optional. If date of first observation is specified, number of day between each incidence observation.      |
| <code>...</code>            | Parameters passed to inner functions  |

### Details

By using different Generation Time (GT) distribution, different estimates of the reproduction ratio can be analyzed.

### Value

A data.frame `s.a` with following components :

|                       |   |
|-----------------------|---|
| <code>GT.type</code>  | Type of distribution for GT.  |
| <code>GT.mean</code>  | Range of means used for tested GTs.   |
| <code>GT.sd</code>    | Range of standard deviations used for tested GTs.   |
| <code>R</code>        | Computed value for Reproduction Number given <code>GT.type</code> , <code>GT.mean</code> and <code>GT.sd</code> . |
| <code>CI.lower</code> | The lower limit of 95% CI for R.  |
| <code>CI.upper</code> | The upper limit of 95% CI for R.  |

### Author(s)

Pierre-Yves Boelle, Thomas Obadia

---

|                      |   |
|----------------------|---|
| <code>sa.time</code> | <i>Sensitivity of R0 to time estimation windows</i> |
|----------------------|---|

---

### Description

Sensitivity analysis to estimate the variation of reproduction numbers according to period over which the incidence is analyzed.

### Usage

```
sa.time(
  incid,
  GT,
  begin = NULL,
  end = NULL,
  est.method,
  t = NULL,
  date.first.obs = NULL,
  time.step = 1,
  res = NULL,
  ...
)
```

### Arguments

|                             |  |
|-----------------------------|--|
| <code>incid</code>          | A vector of incident cases.  |
| <code>GT</code>             | Generation time distribution from <a href="#">generation.time()</a> .  |
| <code>begin</code>          | Vector of begin dates for the estimation of epidemic.  |
| <code>end</code>            | Vector of end dates for estimation of the epidemic.  |
| <code>est.method</code>     | Estimation method used for sensitivity analysis.   |
| <code>t</code>              | Dates vector to be passed to estimation function.  |
| <code>date.first.obs</code> | Optional date of first observation, if <code>t</code> not specified.   |
| <code>time.step</code>      | Optional. If date of first observation is specified, number of day between each incidence observation.   |
| <code>res</code>            | If specified, will extract most of data from a <code>R0.R</code> -class contained within the <code>\$estimate</code> component of a result from <a href="#">estimate.R()</a> and run sensitivity analysis with it. |
| <code>...</code>            | Parameters passed to inner functions   |

### Details

By varying different pairs of begin and end dates, different estimates of reproduction ratio can be analyzed.

'begin' and 'end' vector must have the same length for the sensitivity analysis to run. They can be provided either as dates or numeric values, depending on the other parameters (see `check.incid()`). If some begin/end dates overlap, they are ignored, and corresponding uncomputed data are set to NA.

Also, note that unreliable Rsquared values are achieved for very small time periods (begin ~ end). These values are not representative of the epidemic outbreak behaviour.

### Value

A list with components :

|                       |  |
|-----------------------|--|
| <code>df</code>       | data.frame object with all results from sensitivity analysis.                |
| <code>df.clean</code> | the same object, with NA rows removed. Used only for easy export of results. |
| <code>mat.sen</code>  | Matrix with values of R0 given begin (rows) and end (columns) dates.         |
| <code>begin</code>    | A range of begin dates in epidemic.  |
| <code>end</code>      | A range of end dates in epidemic.  |

### Author(s)

Pierre-Yves Boelle, Thomas Obadia

---

sensitivity.analysis    *Sensitivity analysis of basic reproduction ratio to begin/end dates*

---

### Description

Sensitivity analysis of reproduction ratio using supported estimation methods.

### Usage

```
sensitivity.analysis(
  incid,
  GT = NULL,
  begin = NULL,
  end = NULL,
  est.method = NULL,
  sa.type,
  res = NULL,
  GT.type = NULL,
  GT.mean.range = NULL,
  GT.sd.range = NULL,
  t = NULL,
  date.first.obs = NULL,
  time.step = 1,
  ...
)
```



**Arguments**

|                             |   |
|-----------------------------|---|
| <code>incid</code>          | A vector of incident cases.   |
| <code>GT</code>             | Generation time distribution from <code>generation.time()</code> .  |
| <code>begin</code>          | Vector of begin dates for the estimation of epidemic.   |
| <code>end</code>            | Vector of end dates for estimation of the epidemic.   |
| <code>est.method</code>     | Estimation method used for sensitivity analysis.  |
| <code>sa.type</code>        | String argument to choose between <code>time</code> and <code>GT</code> sensitivity analysis.   |
| <code>res</code>            | If specified, will extract most of data from a <code>R0.R</code> -class result already generated by <code>estimate.R()</code> and run sensitivity analysis on it. |
| <code>GT.type</code>        | Type of distribution for <code>GT</code> (see <code>GT.R</code> for details).   |
| <code>GT.mean.range</code>  | Range of mean values used for all <code>GT</code> distributions throughout the simulation. Must be provided as a vector.  |
| <code>GT.sd.range</code>    | Range of standard deviation values used for <code>GT</code> distributions. Must be provided as a vector.  |
| <code>t</code>              | Dates vector to be passed to estimation function.   |
| <code>date.first.obs</code> | Optional date of first observation, if <code>t</code> not specified.  |
| <code>time.step</code>      | Optional. If date of first observation is specified, number of day between each incidence observation.  |
| <code>...</code>            | Parameters passed to inner functions  |

**Details**

This is a generic call function to use either `sa.time` or `sa.GT`. Argument must be chosen accordingly to `sa.type`. Please refer to `sa.time()` and `sa.GT()` for further details about arguments.

`begin` and `end` vectors must have the same length for the sensitivity analysis to run. They can be provided either as dates or numeric values, depending on the other parameters (see `check.incid()`). If some `begin` or `end` dates overlap, they are ignored and corresponding uncomputed data are set to `NA`. Also, note that unreliable `Rsquared` values are achieved for very small time periods (`begin ~ end`). These values are not representative of the epidemic outbreak behaviour.

**Value**

A sensitivity analysis object of class `R0.S` with components depending on sensitivity analysis `sa.type`.

**Author(s)**

Pierre-Yves Boelle, Thomas Obadia

**Examples**

```
#Loading package
library(R0)

## Data is taken from the paper by Nishiura for key transmission parameters of an institutional
```

```

## outbreak during 1918 influenza pandemic in Germany)
data(Germany.1918)

## For this exemple, we use the exact same call as for the internal sensitivity analysis function

## sa.type = "GT"

## Here we will test GT with means of 1 to 5, each time with SD constant (1)
## GT and SD can be either fixed value or vectors of values
## Actual value in simulations may differ, as they are adapted according to the distribution type
tmp <- sensitivity.analysis(
  sa.type      = "GT",
  incid       = Germany.1918,
  GT.type     = "gamma",
  GT.mean     = seq(1,5,1),
  GT.sd.range = 1,
  begin      = 1,
  end        = 27,
  est.method = "EG")

## Results are stored in a matrix, each line dedicated to a (mean,sd) couple
plot(
  x      = tmp["GT.Mean"],
  xlab  = "mean GT (days)",
  y      = tmp["R"],
  ylim  = c(1.2, 2.1),
  ylab  = "R0 (95% CI)",
  type  = "p",
  pch   = 19,
  col   = "black",
  main  = "Sensitivity of R0 to mean GT"
)
arrows(
  x0 = as.numeric(tmp["GT.Mean"]),
  y0 = as.numeric(tmp["CI.lower"]),
  y1 = as.numeric(tmp["CI.upper"]),
  angle = 90,
  code = 3,
  col = "black",
  length = 0.05
)
## One could tweak this exemple to change sorting of values (per mean, or per standard deviation)
## eg: 'x=tmp[,c('GT.Mean')]' could become 'x=tmp[,c('GT.SD')]'

## sa.type="time"

mGT <- generation.time("gamma", c(2.6,1))
sen <- sensitivity.analysis(
  sa.type      = "time",
  incid       = Germany.1918,
  GT          = mGT,
  begin      = 1:10,

```

```

    end          = 16:25,
    est.method = "EG"
  )
  # ...
  # Warning message:
  # If 'begin' and 'end' overlap, cases where begin >= end are skipped.
  # These cases often return Rsquared = 1 and are thus ignored.
  ## A list with different estimates of reproduction ratio, exponential growth rate and 95%CI
  ## with different pairs of begin and end dates in form of data frame is returned.
  ## If method is "EG", results will include growth rate and deviance R-squared measure
  ## Else, if "ML" method is used, growth rate and R-squared will be set as NA

  ## Interesting results include the variation of R0 given specific begin/end dates.
  ## Such results can be plot as a colored matrix and display Rsquared=f(time period)
  plot(sen, what=c("criterion","heatmap"))
  ## Returns complete data.frame of best R0 value for each time period
  ## (allows for quick visualization)
  ## The "best.fit" is the time period over which the estimate is the more robust

  # $best.fit
  #   Time.period Begin.dates End.dates      R Growth.rate Rsquared CI.lower. CI.upper.
  # 92           15 1970-01-08 1970-01-23 1.64098 0.1478316 0.9752564 1.574953 1.710209

```

---

 sim.epid

*Epidemic outbreak simulation*


---

## Description

Generates several epidemic curves with specified distribution and reproduction number.

## Usage

```

sim.epid(
  epid.nb,
  GT,
  R0,
  epid.length,
  family,
  negbin.size = NULL,
  peak.value = 50
)

```

## Arguments

|         |   |
|---------|---|
| epid.nb | Number of epidemics to be simulated (defaults to 1)                   |
| GT      | Generation time distribution from <a href="#">generation.time()</a> . |
| R0      | Basic reproduction number, in its core definition.                    |

|             |  |
|-------------|--|
| epid.length | Maximum length of the epidemic (cases infected after this length will be truncated).         |
| family      | Distribution of offspring. Can be either "poisson" (default) or "negbin".                    |
| negbin.size | If family is set to "negbin", sets the size parameter of the negative binomial distribution. |
| peak.value  | Threshold value for incidence before epidemics begins decreasing                             |

### Details

This function is only used for simulation purposes. The output is a matrix of n columns (number of outbreaks) by m rows (maximum length of an outbreak).

When using rbinom with mean and size moments, the variance is given by  $\text{mean} + \text{mean}^2/\text{size}$  (see `stats::rbinom()`). One should determine the size accordingly to the  $R_0$  value to increase the dispersion. From the previous formula for the variance,  $\text{Var}(X) = k * R_0$  implies that  $\text{size} = R_0/(k - 1)$ .

### Value

A matrix with epidemics stored as columns (incidence count).

### Author(s)

Pierre-Yves Boelle, Thomas Obadia

### Examples

```
#Loading package
library(R0)

## In this example we simulate n=100 epidemic curves, with peak value at 150 incident cases,
## and maximum epidemic length of 30 time units.
## Only the outbreak phase is computed. When the peak value is reached, the process is stopped
## and another epidemic is generated.
sim.epid(epid.nb=100, GT=generation.time("gamma",c(3,1.5)), R0=1.5,
         epid.length=30, family="poisson", peak.value=150)

# Here, a 30*100 matrix is returned. Each column is a single epidemic.
```

---

sim.epid.indiv      *Influenza-like illness simulation (individual-based model)*

---

### Description

Generates several epidemic curves with an individual-based model.

### Usage

```
sim.epid.indiv(beta, Tmax, n = 1, family = "poisson", negbin.size = NULL)
```

**Arguments**

|             |  |
|-------------|--|
| beta        | Contact rate in the SEIR model.  |
| Tmax        | Maximum length of the epidemic (cases infected after this length will be truncated).         |
| n           | Number of epidemics to be simulated (defaults to 1)  |
| family      | Distribution of offspring. Can be either "poisson" (default) or "negbin".                    |
| negbin.size | If family is set to "negbin", sets the size parameter of the negative binomial distribution. |

**Details**

The epidemic is simulated using a branching process, with infinite number of susceptibles to allow for exponential growth. The model used follows the Crump-Mode-Jagers description, with S/E/I/R description of the natural history.

Latent and infectious period follow parametrized Gamma distributions typical of influenza. An index case is first introduced, and offspring is sampled from a negative binomial distribution, with mean  $\beta * I$  and variance  $negbin.size * \beta * I$ , to allow for overdispersion.

**Value**

A matrix with epidemics stored as columns (incidence count).

**Note**

This is the exact function as used in the manuscript (Obadia et al., 2012).

**Author(s)**

Pierre-Yves Boelle, Thomas Obadia

---

smooth.Rt

*Smooth real-time reproduction number over larger time periods*

---

**Description**

Aggregate real-time estimates of  $R(t)$  (from the TD or even the SB methods) over larger time windows, while still accounting for the generation time distribution.

**Usage**

smooth.Rt(res, time.period)

**Arguments**

|             |   |
|-------------|---|
| res         | An object of class $R0.R$ , created by any real-time method (currently implemented: TD and SB). |
| time.period | Time period to be used for aggregation.   |

## Details

Regrouping Time-Dependant  $R(t)$  values, or even Real Time Bayesian most-likely  $R$  values (according to  $R$  distributions) should take into account the generation time. Results can be plotted exactly the same as input estimations, except they won't show any goodness-of-fit curve.

## Value

A list with components :

|             |   |
|-------------|---|
| R           | The estimate of the reproduction ratio.   |
| conf.int    | The 95% confidence interval for the R estimate.   |
| GT          | Generation time distribution used in the computation.   |
| epid        | Original or augmented epidemic data, depending whether impute.values is set to FALSE or TRUE. |
| begin       | Starting date for the fit.  |
| begin.nb    | The number of the first day used in the fit.  |
| end         | The end date for the fit.   |
| end.nb      | The number of the last day used for the fit.  |
| data.name   | The name of the dataset used.   |
| call        | Call used for the function.   |
| method      | Method used for fitting.  |
| method.code | Internal code used to designate method.   |

## Author(s)

Pierre-Yves Boelle, Thomas Obadia

## Examples

```
#Loading package
library(R0)

## This script allows for generating a new estimation for RTB and TD methods.
## Estimations used as input are aggregated by a time period provided by user.
## Results can be plotted exactly the same as input estimations,
## except they won't show any goodness of fit curve.
data(Germany.1918)
mGT <- generation.time("gamma", c(3,1.5))
TD <- estimate.R(Germany.1918, mGT, begin=1, end=126, methods="TD", nsim=100)
TD
# Reproduction number estimate using Time-Dependant method.
# 2.322239 2.272013 1.998474 1.843703 2.019297 1.867488 1.644993 1.553265 1.553317 1.601317 ...
TD$estimates$TD$Rt.quant
#   Date      R.t. CI.lower. CI.upper.
# 1     1 2.3222391 1.2000000 2.4000000
# 2     2 2.2720131 2.7500000 6.2500000
# 3     3 1.9984738 2.7500000 6.5000000
```

```

# 4      4 1.8437031 0.7368421 1.5789474
# 5      5 2.0192967 3.1666667 6.1666667
# 6      6 1.8674878 1.6923077 3.2307692
# 7      7 1.6449928 0.8928571 1.6428571
# 8      8 1.5532654 1.3043478 2.2608696
# 9      9 1.5533172 1.0571429 1.7428571
# 10     10 1.6013169 1.6666667 2.6666667
# ...

```

```
TD.weekly <- smooth.Rt(TD$estimates$TD, 7)
```

```
TD.weekly
```

```
# Reproduction number estimate using Time-Dependant method.
```

```
# 1.878424 1.580976 1.356918 1.131633 0.9615463 0.8118902 0.8045254 0.8395747 0.8542518 0.8258094..
```

```
TD.weekly$Rt.quant
```

```

#   Date      R.t. CI.lower. CI.upper.
# 1     1 1.8784240 1.3571429 2.7380952
# 2     8 1.5809756 1.3311037 2.0100334
# 3    15 1.3569175 1.1700628 1.5308219
# 4    22 1.1316335 0.9961229 1.2445302
# 5    29 0.9615463 0.8365561 1.0453074
# 6    36 0.8118902 0.7132668 0.9365193
# 7    43 0.8045254 0.6596685 0.9325967
# 8    50 0.8395747 0.6776557 1.0402930
# 9    57 0.8542518 0.6490251 1.1086351
# 10   64 0.8258094 0.5836735 1.1142857
# 11   71 0.8543877 0.5224719 1.1460674
# 12   78 0.9776385 0.6228070 1.4912281
# 13   85 0.9517133 0.5304348 1.3652174
# 14   92 0.9272833 0.5045045 1.3423423
# 15   99 0.9635479 0.4875000 1.5125000
# 16  106 0.9508951 0.5000000 1.6670455
# 17  113 0.9827432 0.5281989 1.8122157
# 18  120 0.5843895 0.1103040 0.9490928

```

# Index

## \* datasets

Germany.1918, 8  
GT.chld.hsld, 8  
H1N1.serial.interval, 9

as.Date(), 13

censored.deviation(), 10  
check.incid(), 10–13, 16, 17

est.GT, 2  
est.R0.AR(), 4, 5  
est.R0.EG(), 5  
est.R0.ML(), 5  
est.R0.SB(), 5  
est.R0.TD(), 5  
estimate.R, 3  
estimate.R(), 15, 17

generation.time, 6  
generation.time(), 2–4, 11, 12, 15, 17, 19  
Germany.1918, 8  
GT.chld.hsld, 8

H1N1.serial.interval, 9

impute.incid, 10  
inspect.data, 10  
inspect.data(), 11  
integrity.checks, 12  
integrity.checks(), 4, 12

sa.GT, 13  
sa.GT(), 17  
sa.time, 15  
sa.time(), 17  
sensitivity.analysis, 16  
sim.epid, 19  
sim.epid.indiv, 20  
smooth.Rt, 21  
stats::rnbino(), 20