

Package ‘pestim’

March 8, 2018

Type Package

Title Population Estimations Using Mobile Phone Data

Version 0.1.0

Date 2018-02-01

Description This package contains functions that implement a simple hierarchical model to estimate the population counts of different territorial cells combining the information from aggregated mobile phone data and a population register or survey data.

License GPL-3 and EUPL

Encoding UTF-8

LazyData true

Depends R (>= 3.3.0)

Imports data.table (>= 1.10.4),
Rcpp (>= 0.12.12),
MCMCpack (>= 1.4-2),
RcppParallel

SystemRequirements GNU make

LinkingTo Rcpp, RcppParallel

RoxygenNote 6.0.1

Suggests knitr, rmarkdown

VignetteBuilder knitr

Collate 'MobPop.R'
'ratioBeta.R'
'kummer.R'
'Phi.R'
'RcppExports.R'
'alphaPrior.R'
'data.R'
'dg.R'
'triang.R'
'dlambda.R'
'flambda.R'
'fu.R'
'fv.R'
'genAlpha.R'

'genUV.R'
 'modeLambda.R'
 'nMNO_ini.R'
 'nReg.R'
 'pestim.R'
 'pkummer.R'
 'rlambda.R'
 'rN0.R'
 'postN0.R'
 'rmatProb.R'
 'rNtcondN0.R'
 'rNt.R'
 'postNt.R'
 'postNtcondN0.R'
 'rg.R'
 'rp.R'

R topics documented:

alphaPrior	3
dg	4
dlambda	5
flambda	7
fu	7
fv	8
genAlpha	8
genUV	9
kummer	10
MobPop	11
modeLambda	11
nMNO_ini	13
nReg	13
pestim	13
Phi	15
pkummer	16
postN0	16
postNt	18
postNtcondN0	20
ratioBeta	21
rg	21
rlambda	22
rmatProb	24
rN0	25
rNt	26
rNtcondN0	28
rp	30
triang	31

alphaPrior	<i>Generate prior distributions for parameters of the Dirichlet distribution.</i>
------------	---

Description

Generate a list of prior distributions for the parameters of the Dirichlet distribution in the hierarchical model. Each component of the list corresponds to the prior distribution of the parameter $\alpha_{ij}(t_0, t_n)$ for each cell j . This function initial works over a fixed initial cell i . Each returned distribution is specified as a list with an identification name as first component and named components with the distribution parameters for the rest of components.

Usage

```
alphaPrior(nMNOfrom, names, variation)
```

Arguments

nMNOfrom	numeric vector with the number of individuals moving from the initial cell to the rest of cells (including those remaining)
names	character vector with the names of the prior distributions for each cell
variation	list of lists whose components are parameters providing a measure of variation of each prior distribution

Details

The function takes the number of cells from the input parameter nMNOfrom which specifies the number of individuals detected by the network moving from the initial cell to each of the cells (including those remaining in the same). The function executes the same construction for each final cell. It takes the name of prior distribution from the input parameter names and construct the corresponding prior distribution for each cell j with mode at $u_j^* = N_j$, where N_j is taken from nMNOfrom. Next the rest of parameters of the distribution are computed according to the dispersion parameters specified in variation.

As accepted distribution names, currently the user can specify unif, degen, triang, and gamma.

The dispersion parameters recognised so far are the coefficients of variation only (standard deviation divided by the mean of the distribution). These dispersion parameters must be specified by a named component cv with a numeric value in $[0, 1]$.

For each distribution the parameters are computed as follows:

- **unif**: This is the uniform distribution with parameters xMax and xMin. Both parameters are computed by $u_j^* \cdot (1 \pm \sqrt{3}cv)$, respectively, in each cell j .
- **degen**: This is the degenerate distribution with parameter X_0 taken as u_j^* in each cell j .
- **triang**: This is the triangular distribution **dtriang** with parameters xMax, xMin, and xMode. The latter is taken directly from nMNOfrom. The distribution is assumed to be symmetrical so that the two former parameters are computed by $u_j^* \cdot (1 \pm \sqrt{3}cv)$, respectively, in each cell j .
- **gamma**: This is the gamma distribution with parameters shape and scale. The former is computed as $\frac{1}{cv^2}$ and the latter as $frac{u_j^*}{scale} - 1$.

Value

Return a list with a list in each component specifying the prior for each cell

Examples

```
# Three cells. Cell 1 under study. 10 individuals remain.
alphaPrior(c(10, 3, 4), c('unif', 'triang', 'gamma'),
           list(list(cv = 0.1), list(cv = 0.05), list(cv = 0.15)))
```

dg	<i>Density function of a candidate distribution in the accept-reject method.</i>
----	--

Description

Generate values of a candidate distribution density function in the accept-reject method of generation of random variables applied to the distribution of the lambda parameter

Usage

```
dg(lambda, nMNO, nReg, fu, fv, flambda, relTol = 1e-06, nSim = 10000,
    nStrata = c(1, 100), verbose = FALSE)
```

Arguments

lambda	numeric vector with the lambda parameter values
nMNO,	nReg non-negative integer vectors with the number of individuals detected in each cell according to the network operator and the register
fu,	fv named lists with the prior marginal distributions of the two-dimensional points for the Monte Carlo integration
flambda	named list with the prior distribution of the lambda parameter
relTol	relative tolerance in the computation of the kummer function. Default value is 1e-6
nSim	number of two-dimensional points to generate to compute the integral. Default value is 1e4
nStrata	integer vector of length 2 with the number of strata in each dimension. Default value is c(1, 1e2)
verbose	logical (default FALSE) to report progress of the computation

Details

The candidate distribution is a gamma distribution with parameters shape = nMNO + 1 and scale = λ^* / nMNO, where λ^* stands for the mode of the posterior distribution of the lambda parameter.

It is important to know that currently this function accepts only parameters for a single cell at a time. In case of interest for the candidate density function values for a set of cells, the user should program his/her own routine to apply this function to every cell.

Value

dg generates `length(lambda)` values of the density probability function of the candidate distribution in the accept-reject method.

See Also

[modelLambda](#), [dlambda](#) for related functions.

Examples

```
curve(dg(x, nMNO = 20, nReg = 115, fu = list('unif', xMin = 0.3, xMax = 0.5),
      fv = list('unif', xMin = 100, xMax = 120),
      flambda = list('gamma', shape = 11, scale = 12)), xlim = c(0, 150),
      main = '', ylab = 'density', xlab = 'lambda')
```

dlambda	<i>Posterior density function of the lambda parameter.</i>
---------	--

Description

Compute the unnormalized posterior density function of the parameter λ in the hierarchical model to estimate population counts given by

$$f(\lambda | N^{\text{MNO}}; N^{\text{Nreg}}) \propto f(\lambda) \cdot \text{dpois}(N^{\text{MNO}}; \lambda) \cdot S(\lambda; N^{\text{MNO}}, N^{\text{Nreg}}),$$

where [dpois](#) is the probability density function of a Poisson distribution and S is defined in the bibliographic reference.

Usage

```
dlambda(lambda, nMNO, nReg, fu, fv, flambda, relTol = 1e-06, nSim = 1000,
        nStrata = c(1, 100), verbose = FALSE)
```

Arguments

lambda	numeric vector
nMNO,	nReg non-negative integer vectors with the number of individuals detected in each cell according to the network operator and the register
fu,	fv named lists with the prior marginal distributions of the two-dimensional points for the Monte Carlo integration
flambda	named list with the prior distribution of the lambda parameter
relTol	relative tolerance in the computation of the kummer function. Default value is 1e-6
nSim	number of two-dimensional points to generate to compute the integral. Default value is 1e3
nStrata	integer vector of length 2 with the number of strata in each dimension. Default value is c(1, 1e2)
verbose	logical (default FALSE) to report progress of the computation

Details

The lengths of the input vectors `nMNO` and `nReg` must be both equal to 1 and independent of the length of the input vector `lambda`. The integral is computed using with Monte Carlo techniques using `nSim` points for each of the values `lambda` specified so that the final `data.table` has `length(lambda)` rows.

The prior distributions are specified as named lists where the first component of each list must be the name of distribution (`'unif'`, `'triang'`, `'degen'`, `'gamma'`) and the rest components must be named according to the name of the parameters of the random generator of the corresponding distribution according to:

- `unif`: `xMin`, `xMax` for the minimum, maximum of the sampled interval.
- `degen`: `x0` for the degenerate value of the random variable.
- `triang`: `xMin`, `xMax`, `xMode` for minimum, maximum and mode (see [qtriang](#)).
- `gamma`: `scale` and `shape` with the same meaning as in [rgamma](#).

It is important to know that currently this function accepts only parameters for a single cell at a time. In case of interest for the density function values for a set of cells, the user should program his/her own routine to apply this function to every cell.

Value

`dlambda` returns a `data.table` with the values of the density function (column `probLambda`) for each value of `lambda` together with additional variables:

- The common length of `nMNO` and `nReg` identifies the number of territorial cells in which the number of individuals detected by the telecommunication network and official data. The column `cellID` identifies these territorial cells.
- The length of `lambda` identifies the number of parameters upon which the integral will be computed in each cell. The column `parID` identifies each of these input parameters.
- The inputs `nMNO` and `nReg` are also included in the output `data.table` in columns under the same name.
- The value on the integral times the Poisson density function if also included under the column `integral`

References

<https://github.com/MobilePhoneESSnetBigData>

See Also

[genUV](#), [Phi](#) for related functions.

Examples

```
# This data.table must have 5x3= 15 rows
dlambda(seq(0, 1, length.out = 5),
        nMNO = c(20, 17, 25), nReg = c(115, 123, 119),
        fu = list('unif', xMin = 0.3, xMax = 0.5), fv = list('gamma', shape = 11, scale = 12),
        flambda = list('gamma', shape = 11, scale = 12))

# Easily, a function to draw conditioned on the parameters:
f <- function(x){
  dlambda(x, nMNO = 20, nReg = 115,
```

```

fu = list('unif', xMin = 0.3, xMax = 0.5), fv = list('unif', xMin = 100, xMax = 120),
  flambda = list('gamma', shape = 11, scale = 12))$probLambda
}
curve(f, xlim = c(0, 150))

```

flambda

*List of priors for the parameter lambda for the dataset MobPop.***Description**

This list contains the priors for each of the 12 cells of the simulated population included in the [data.table](#) MobPop.

Usage

flambda

Format

A list with 12 components each of which is a list with three components:

name of the prior distribution (gamma in all cases in this example)

xMin shape parameter for the gamma prior distribution of each cell

xMax scale parameter for the gamma prior distribution of each cell

fu

*List of priors for the parameter u for the dataset MobPop.***Description**

This list contains the priors for each of the 12 cells of the simulated population included in the [data.table](#) MobPop (see function [genUV](#)).

Usage

fu

Format

A list with 12 components each of which is a list with three components:

name of the prior distribution (unif in all cases in this example)

xMin minimum value of the range of values of the uniform prior distribution of each cell

xMax maximum value of the range of values of the uniform prior distribution of each cell

fv	<i>List of priors for the parameter v for the dataset MobPop.</i>
----	---

Description

This list contains the priors for each of the 12 cells of the simulated population included in the [data.table](#) MobPop (see function [genUV](#)).

Usage

fv

Format

A list with 12 components each of which is a list with three components:

name of the prior distribution (unif in all cases in this example)

xMin minimum value of the range of values of the uniform prior distribution of each cell

xMax maximum value of the range of values of the uniform prior distribution of each cell

genAlpha	<i>Generate values for the parameters of the Dirichlet distribution.</i>
----------	--

Description

Generate a matrix of values of the parameters $\alpha_{ij}(t_0, t_n)$ of the Dirichlet distribution in the hierarchical model. This function initially works over a fixed initial cell i under study.

Usage

```
genAlpha(nSim, flist)
```

Arguments

nSim	number of values to generate
flist	list with the prior distributions for each cell

Details

This function generates the nSim random values according to the prior of each cell specified in flist.

The prior distributions are specified as named lists where the first component of each list must be the name of distribution ('unif', 'triang', 'degen', 'gamma') and the rest of components must be named according to the name of the parameters of the random generator of the corresponding distribution according to:

- unif: xMin, xMax for the minimum, maximum of the sampled interval.
- degen: x0 for the degenerate value of the random variable.
- triang: xMin, xMax, xMode for minimum, maximum and mode (see [qtriang](#)).
- gamma: scale and shape with the same meaning as in [rgamma](#).

Value

Return a matrix with as many columns as cells and as many rows as number of generated values

Examples

```
priors <- alphaPrior(c(10, 3, 4), c('unif', 'triang', 'gamma'),
                     list(list(cv = 0.1), list(cv = 0.05), list(cv = 0.15)))
genAlpha(10, priors)
```

genUV	<i>Generation of two-dimensional random deviates.</i>
-------	---

Description

Generate two-dimensional random deviates for a Monte Carlo computation of the integral

$$\int_0^\infty dv f_2(v) \int_0^\infty f_1(u) \Phi(u \cdot v, (1-u) \cdot v; \lambda, N^{\text{MNO}}, N^{\text{Reg}}).$$

The Monte Carlo technique makes use of stratified importance sampling.

Usage

```
genUV(nSim, nStrata, f1, f2, lambda, nMNO, nReg)
```

Arguments

nSim	number of two-dimensional points to generate
nStrata	integer vector of length 2 with the number of strata in each dimension
f1,	f2 named lists with the prior marginal distributions of the two-dimensional points
lambda	numeric vector
nMNO,	nReg non-negative integer vectors

Details

The lengths of the input vectors nMNO and nReg must be equal and independent of the length of the input vector lambda. Notice that nSim points are generated for each of the $\text{length}(\text{nMNO}) \times \text{length}(\text{lambda})$ combinations so that the final [data.table](#) has $\text{nSim} \times \text{length}(\text{nMNO}) \times \text{length}(\text{lambda})$ rows.

The prior distributions are specified as named lists where the first component of each list must be the name of distribution ('unif', 'triang', 'degen', 'gamma') and the rest components must be named according to the name of the parameters of the random generator of the corresponding distribution according to:

- unif: xMin, xMax for the minimum, maximum of the sampled interval.
- degen: x0 for the degenerate value of the random variable.
- triang: xMin, xMax, xMode for minimum, maximum and mode (see [qtriang](#)).
- gamma: scale and shape with the same meaning as in [rgamma](#).

Value

genUV returns a [data.table](#) with the (u,v) coordinates of each point together with additional variables:

- The common length of nMNO and nReg identifies the number of territorial cells in which the number of individuals detected by the telecommunication network and official data. The column cellID identifies these territorial cells.
- The length of lambda identifies the number of parameters upon which the integral will be computed in each cell. The column parID identifies each of these input parameters.
- Stratum_u and Stratum_v jointly identify each stratum in which the region of integration has been divided with the stratification.

See Also

[runif](#), [qtriang](#), [rgamma](#) for related functions.

Examples

```
# This data.table must have 10x5x3= 150 rows and only one stratum
genUV(nSim = 10, nStrata = c(1, 1),
      f1 = list('unif', xMin = 0.3, xMax = 0.5), f2 = list('gamma', shape = 11, scale = 12),
      lambda = seq(0, 1, length.out = 5),
      nMNO = c(20, 17, 25), nReg = c(115, 123, 119))
```

kummer

Confluent hypergeometric or Kummer function

Description

Partial implementation of the confluent hypergeometric function ${}_1F_1(x; a; b)$

Usage

```
kummer(x, a, b, relTol = 1e-06)
```

Arguments

x,	a, b numeric vectors of the same length
relTol	relative tolerance (default value 1e-6) understood as the ratio of each term in the series relative to the sum

Details

This function is implemented in C++. It is based on Pearson et al (2016). It only implements the Taylor series method together with an asymptotic expansion based on Watson's lemma

Value

Return a numeric vector with the values of the function

Author(s)

Luis Sanguiao Bogdan Oancea

MobPop

Dataset with simulated data for population counts.

Description

This dataset provides population counts moving from each pair of cells at successive time instants for a simulated true population, a corresponding official population in a register and a population detected with a mobile telecommunication network.

This dataset provides population counts moving from each pair of cells at successive time instants for a simulated true population, a corresponding official population in a register and a population detected with a mobile telecommunication network.

Usage

```
MobPop
```

```
MobPop
```

Format

A [data.table](#) with 96768 rows and 6 variables:

ID_CELL_INI identification code for each initial cell in the displacements

ID_CELL_END identification code for each final cell in the displacements

ID_T identification code of each time moment. It is very important to underline that the table collects always displacements between the initial time instant and the corresponding time instant specified by ID_T

N_REG counts according to the population register. Note that these counts do not evolve in time

N_0 counts of the simulated true population

N_MNO_1 counts of individuals detected by the Mobile Network Operator

modeLambda

Mode of the posterior density function of the lambda parameter.

Description

Compute the mode of the unnormalized posterior density function of the parameter λ in the hierarchical model to estimate population counts.

Usage

```
modeLambda(nMNO, nReg, fu, fv, flambda, relTol = 1e-06, nSim = 10000,
  nStrata = c(1, 100), verbose = FALSE)
```

Arguments

nMNO,	nReg non-negative integer vectors with the number of individuals detected in each cell according to the network operator and the register
fu,	fv named lists with the prior marginal distributions of the two-dimensional points for the Monte Carlo integration
flambda	named list with the prior distribution of the lambda parameter
relTol	relative tolerance in the computation of the kummer function. Default value is 1e-6
nSim	number of two-dimensional points to generate to compute the integral. Default value is 1e4
nStrata	integer vector of length 2 with the number of strata in each dimension. Default value is c(1, 1e2)
verbose	logical (default FALSE) to report progress of the computation

Details

The lengths of the input vectors nMNO and nReg must be equal. Currently the optimization algorithm is a simple direct algorithm taking into account the form of the density function.

The prior distributions are specified as named lists where the first component of each list must be the name of distribution ('unif', 'triang', 'degen', 'gamma') and the rest components must be named according to the name of the parameters of the random generator of the corresponding distribution according to:

- unif: xMin, xMax for the minimum, maximum of the sampled interval.
- degen: x0 for the degenerate value of the random variable.
- triang: xMin, xMax, xMode for minimum, maximum and mode (see [qtriang](#)).
- gamma: scale and shape with the same meaning as in [rgamma](#).

Value

modeLambda returns a vector with the values of the mode of the density function (column probLambda) for each cell.

See Also

[dlambda](#) for the function to maximize.

Examples

```
# This data.table must have 5x3= 15 rows
modeLambda(nMNO = c(20, 17, 25), nReg = c(115, 123, 119),
  fu = list(list('unif', xMin = 0.3, xMax = 0.5),
    list('unif', xMin = 0.35, xMax = 0.45),
    list('unif', xMin = 0.25, xMax = 0.43)),
  fv = list(list('gamma', shape = 11, scale = 12),
    list('gamma', shape = 12, scale = 12.3),
    list('gamma', shape = 13, scale = 11.5)),
  flambda = list(list('gamma', shape = 11, scale = 12),
    list('gamma', shape = 12, scale = 12.3),
    list('gamma', shape = 13, scale = 12)))
```

nMNO_ini	<i>Counts of individuals for the initial time period detected by the mobile network operator.</i>
----------	---

Description

This vector contains the counts of individuals for the initial time period detected by the mobile network operator in each of the 12 cells.

Usage

```
nMNO_ini
```

Format

A vector with 12 components.

nReg	<i>Population counts for the initial time period according to the population register.</i>
------	--

Description

This vector contains the population counts for the initial time period according to the population register.

Usage

```
nReg
```

Format

A vector with 12 components.

pestim	<i>pestim: a hierarchical model to estimate population counts with aggregated mobile phone data.</i>
--------	--

Description

This package provides an implementation for a hierarchical model to combine both aggregated mobile phone data and external official (administrative or survey) data to produce estimates of population counts in each cell of a division of a territory.

Context

This package has been developed in the context of a European research project within the European Statistical System called **ESSnet on Big Data**. More specifically this work corresponds to the work package on mobile phone data by which we assess the use of this data source in the production of official statistics. The goals of the project is many-fold. Firstly, the issue of accessing these data for the production of official statistics initially for research and then for standard production has been investigated. Secondly, in a hands-on bottom-up approach, we make some initial methodological proposals to produce concrete statistical output using those data sets compiled in the preceding phase. Thirdly, in parallel, IT tools, architecture and software development are assessed especially in contrast to traditional computer frameworks. Finally, quality is appraised especially in the context of the European Statistics Code of Practice and ESS Quality Assurance Framework. This package provides a first-step implementation of software routines to present a proof of concept about a methodological proposal (see below) to make inferences about a target population from a mobile phone dataset.

The hierarchical model in a nutshell

The methodological proposal giving rise to this package focuses on the inference exercise connecting aggregated mobile phone data with a target population under analysis. In concrete, the goal is to provide estimates of population counts in each cell in which we have divided the territory for which the telecommunication network provides count data. The estimation is assisted with official data at a larger time scale (either from a population register or from a survey).

The model rests on two working assumptions:

- Given that mobile phone data and official data operate at different time scales, we assume that there exists an initial time instant in which we can equate population figures from both sources.
- The mobility patterns of individuals do not depend on the mobile network operator which they are subscribed to.

The model works in two stages. Firstly at the initial time instant, we use data from both sources to make the inference for the actual population counts in each cell. Secondly, the time evolution of these counts are produced using the transition matrices from cell to cell of individuals provided by the mobile network operator.

The essence of the model is to emulate the ecological sampling setting in which the number of detected individuals in each cell follows a binomial distribution $Bin(N_i, p_i)$ whose parameter N_i is the target of the model and is assigned a weakly informative prior and the detection probability is also assigned a weakly informative prior based upon both data sources.

Computational paradigm

Computations are conducted following the Bayesian paradigm. In this sense the generation of simulated populations according to different probability distributions is at the core of the package. In this sense the package contains basically three types of functions:

- Auxiliary functions, providing computation of mathematical functions such as the ratio of two beta functions, the confluent hypergeometric function, an optimization routine for a concrete probability distribution, etc. Examples of these functions are [ratioBeta](#), [kummer](#), [Phi](#), [modelLambda](#).
- Distribution-relation functions, providing computation regarding the generation of random deviates according to different probability distributions comprising both priors, posteriors, and

the generation of parameter specifications for these distributions. Examples of these functions are [dtriang](#), [rtriang](#), [ptriang](#), [qtriang](#), [dlambda](#), [rlambda](#), [rmatProb](#), [rN0](#), [rNt](#), [rNtcondN0](#), [rg](#), [rp](#), [alphaPrior](#), [genAlpha](#), [genUV](#).

- Estimation-relation functions, providing computation of estimates based upon the populations generated with the preceding functions. Examples of these functions are [postN0](#), [postNt](#), [postNtcondN0](#).

Phi

The product of ratioBeta and Kummer functions

Description

Compute the product of [ratioBeta](#) and [kummer](#) functions with a specific set of arguments

Usage

```
Phi(alpha, beta, lambda, n, relTol = 1e-06)
```

Arguments

alpha,	beta non-negative numeric vectors
lambda	numeric vector
n	non-negative integer vector
relTol	relative tolerance in the computation of the kummer function. Default value is 1e-6

Value

Phi returns $\frac{B(\alpha+m, \beta+n)}{B(\alpha, \beta)} \cdot {}_1F_1(\lambda; \alpha; \beta)$, where ${}_1F_1$ stands for the confluent hypergeometric function

The lengths of the input vectors must be all equal except when their length is 1, which are recycled. Otherwise NAs are produced.

See Also

[ratioBeta](#), [kummer](#) for related functions.

Examples

```
Phi(1, 1, 0.5, 10)
Phi(1:10, 10:1, seq(0, 1, length.out = 10), 3)
Phi(1:4, 4:1, c(2, 3), c(4, 3, 1))
```

pkummer

*Confluent hypergeometric or Kummer function***Description**

A parallel implementation of the confluent hypergeometric function ${}_1F_1(x; a; b)$ using RcppParallel package

Usage

```
pkummer(x, a, b, relTol = 1e-06)
```

Arguments

x,	a, b numeric vectors of the same length
relTol	relative tolerance (default value 1e-6) understood as the ratio of each term in the series relative to the sum

Details

This function is implemented in C++. It is based on Pearson et al (2016). It implements the Taylor series method together with an asymptotic expansion based on Watson's lemma. The actual implementation uses parallelFor from RcppParallel package together with a functor that computes confluent hypergeometric function for chunks of vectors x, a, b

Value

Return a numeric vector with the values of the function

Author(s)

Bogdan Oancea

postN0

*Posterior mean, median, and mode for the number of individuals at the initial time.***Description**

Compute the posterior mean, median, and mode for the number of individuals generating posterior distribution according to the hierarchical model at the initial time instant

Usage

```
postN0(nMNO, nReg, fu, fv, flambda, n = 1000, scale = 1, relTol = 1e-08,
       nSim = 1000, nStrata = c(1, 100), verbose = FALSE)
```


Arguments

nMNO,	nReg non-negative integer vectors with the number of individuals detected in each cell according to the network operator and the register
fu,	fv named lists with the prior marginal distributions of the two-dimensional points for the Monte Carlo integration
flambda	named list with the prior distribution of the lambda parameter
n	number of points to generate in the posterior distribution for the computation. Default value is 1e3
scale	numeric vector with the scale to count the number of individuals. Default value is 1
relTol	relative tolerance in the computation of the kummer function. Default value is 1e-6
nSim	number of two-dimensional points to generate to compute the integral. Default value is 1e4
nStrata	integer vector of length 2 with the number of strata in each dimension. Default value is c(1, 1e2)
verbose	logical (default FALSE) to report progress of the computation

Details

The prior distributions are specified as named lists where the first component of each list must be the name of distribution ('unif', 'triang', 'degen', 'gamma') and the rest of components must be named according to the name of the parameters of the random generator of the corresponding distribution according to:

- unif: xMin, xMax for the minimum, maximum of the sampled interval.
- degen: x0 for the degenerate value of the random variable.
- triang: xMin, xMax, xMode for minimum, maximum and mode (see [qtriang](#)).
- gamma: scale and shape with the same meaning as in [rgamma](#).

Value

postN0 computes the posterior mean, median, and mode of the posterior distribution for each cell. The function returns a matrix with the estimates in columns and the cells in rows.

See Also

[rN0](#)

Examples

```
# It takes a couple of minutes
postN0(nMNO = 20, nReg = 115, fu = list('unif', xMin = 0.3, xMax = 0.5),
      fv = list('unif', xMin = 100, xMax = 120),
      flambda = list('gamma', shape = 11, scale = 12))
```

postNt	<i>Posterior mean, median, and mode for the number of individuals at an arbitrary time.</i>
--------	---

Description

Compute the posterior mean, median, and mode for the number of individuals generating posterior distribution according to the hierarchical model.

Usage

```
postNt(nMNOmat, nReg, fu, fv, flambda, distNames, variation, scale = 1,
       n = 1000, relTol = 1e-06, nSim = 1000, nStrata = c(1, 100),
       verbose = FALSE)
```

Arguments

nMNOmat	transition matrix with the number of individuals displaced from cell to cell detected by the Mobile Network Operator
nReg	non-negative integer vector with the number of individuals detected in each cell according to the population register
fu,	fv named lists with the prior marginal distributions of the two-dimensional points for the Monte Carlo integration
flambda	named list with the prior distribution of the lambda parameter
distNames	character vector with the names of the prior distributions for each cell
variation	list of lists whose components are parameters providing a measure of variation of each prior distribution
scale	numeric vector with the scale to count the number of individuals. Default value is 1
n	number of points to generate in the posterior distribution for the computation. Default value is 1e3
relTol	relative tolerance in the computation of the kummer function. Default value is 1e-6
nSim	number of two-dimensional points to generate to compute the integral. Default value is 1e4
nStrata	integer vector of length 2 with the number of strata in each dimension. Default value is c(1, 1e2)
verbose	logical (default FALSE) to report progress of the computation

Details

The prior distributions are specified as named lists where the first component of each list must be the name of distribution ('unif', 'triang', 'degen', 'gamma') and the rest of components must be named according to the name of the parameters of the random generator of the corresponding distribution according to:

- unif: xMin, xMax for the minimum, maximum of the sampled interval.
- degen: x0 for the degenerate value of the random variable.
- triang: xMin, xMax, xMode for minimum, maximum and mode (see [qtriang](#)).
- gamma: scale and shape with the same meaning as in [rgamma](#).

Value

postNt computes the posterior mean, median, and mode of the posterior distribution for each cell at an arbitrary time t . The function returns a matrix with the estimates in columns and the cells in rows.

See Also

[rNt](#), [postN0](#), [postNtcondN0](#)

Examples

```
## First, the inputs:

#The transition matrix of individuals detected by the MNO
nMNOmat <- rbind(c(10, 3, 4), c(5, 21, 3), c(3, 9, 18))

# Population at the initial time of each cell according to the population register
nReg <- c(90, 130, 101)

# List of priors for u
u0 <- rowSums(nMNOmat) / nReg
cv_u0 <- 0.15
fu <- lapply(u0, function(u){
  umin <- max(0, u - cv_u0 * u)
  umax <- min(1, u + cv_u0 * u)
  output <- list('unif', xMin = umin, xMax = umax)
  return(output)
})

# List of priors for v
v0 <- nReg
cv_v0 <- 0.10
fv <- lapply(v0, function(u){
  umin <- max(0, u - cv_v0 * u)
  umax <- u + cv_v0 * u
  output <- list('unif', xMin = umin, xMax = umax)
  return(output)
})

# List of priors for lambda
cv_lambda <- 0.6
alpha <- 1 / cv_lambda**2 - 1
flambda <- lapply(v0, function(v){list('gamma', shape = 1 + alpha, scale = v / alpha)})

# Names and parameters of priors for the transition probabilities
distNames <- rep('unif', 3)
variation <- rep(list(list(cv = 0.20)), 3)

# It takes a couple of minutes.
postNt(nMNOmat, nReg, fu, fv, flambda, distNames, variation)
```

postNtcondN0	<i>Posterior mean, median, and mode for the number of individuals at an arbitrary time conditioned upon the initial population.</i>
--------------	---

Description

Compute the posterior mean, median, and mode for the number of individuals generating posterior distribution according to the hierarchical model conditioned upon the initial population of each cell, which must be provided

Usage

```
postNtcondN0(N0, nMNOmat, distNames, variation, n = 1000)
```

Arguments

N0	initial population in each cell
nMNOmat	transition matrix with the number of individuals displaced from cell to cell detected by the Mobile Network Operator
distNames	character vector with the names of the prior distributions for each cell
variation	list of lists whose components are parameters providing a measure of variation of each prior distribution
n	number of points to generate in the posterior distribution for the computation. Default value is 1e3

Value

Return a matrix with three columns (mean, median, and mode estimates) and one row per cell

Examples

```
## First, the inputs:

# The initial population
N0 <- c(93, 123, 130)

#The transition matrix of individuals detected by the MNO
nMNOmat <- rbind(c(10, 3, 4), c(5, 21, 3), c(3, 9, 18))

# Names and parameters of priors for the transition probabilities
distNames <- rep('unif', 3)
variation <- rep(list(list(cv = 0.20)), 3)

# It takes a couple of minutes.
postNtcondN0(N0, nMNOmat, distNames, variation)
```

ratioBeta	<i>The ratio of two beta functions.</i>
-----------	---

Description

Compute the ratio of two beta functions whose arguments differ by integer numbers

Usage

```
ratioBeta(alpha, beta, m, n)
```

Arguments

alpha,	beta non-negative numeric vectors
m,	n non-negative integer vectors

Value

ratioBeta gives $\frac{B(\alpha+m, \beta+n)}{B(\alpha, \beta)}$

The lengths of the input vectors must be all equal except when their length is 1, which are recycled. Otherwise NAs are produced.

See Also

[beta](#), [lbeta](#) for related functions.

Examples

```
ratioBeta(10, 13, 2, 3)
ratioBeta(1:10, 10:1, 2, 3)
ratioBeta(1:3, 3:1, c(2, 3), 4)
```

rg	<i>Generation of random deviates of the candidate distribution.</i>
----	---

Description

Generate random points according to the candidate probability distribution in the accept-reject method of generation of random variables applied to the distribution of the lambda parameter

Usage

```
rg(n, nMNO, nReg, fu, fv, flambda, relTol = 1e-06, nSim = 10000,
   nStrata = c(1, 100), verbose = FALSE)
```

Arguments

n	number of values to generate
nMNO,	nReg non-negative integer vectors with the number of individuals detected in each cell according to the network operator and the register
fu,	fv named lists with the prior marginal distributions of the two-dimensional points for the Monte Carlo integration
flambda	named list with the prior distribution of the lambda parameter
relTol	relative tolerance in the computation of the kummer function. Default value is 1e-6
nSim	number of two-dimensional points to generate to compute the integral. Default value is 1e4
nStrata	integer vector of length 2 with the number of strata in each dimension. Default value is c(1, 1e2)
verbose	logical (default FALSE) to report progress of the computation

Details

The candidate distribution is a gamma distribution with parameters $\text{shape} = \text{nMNO} + 1$ and $\text{scale} = \lambda^* / \text{nMNO}$, where λ^* stands for the mode of the posterior distribution of the lambda parameter.

It is important to know that currently this function accepts only parameters for a single cell at a time. In case of interest for the candidate density function values for a set of cells, the user should program his/her own routine to apply this function to every cell.

Value

rg generates n points according to the candidate distribution.

See Also

[modelLambda](#), [dlambda](#) for related functions.

Examples

```
hist(rg(1e5, nMNO = 20, nReg = 115, fu = list('unif', xMin = 0.3, xMax = 0.5),
      fv = list('unif', xMin = 100, xMax = 120),
      flambda = list('gamma', shape = 11, scale = 12)), breaks = seq(1, 200, by = 2), main = '')
```

rlambda

Generation of random deviates of the posterior distribution of parameter lambda.

Description

Generate random points according to the posterior probability distribution of the parameter lambda in the hierarchical model.

Usage

```
rlambda(n, nMNO, nReg, fu, fv, flambda, relTol = 1e-06, nSim = 10000,
        nStrata = c(1, 100), verbose = FALSE)
```

Arguments

n	number of values to generate
nMNO,	nReg non-negative integer vectors with the number of individuals detected in each cell according to the network operator and the register
fu,	fv named lists with the prior marginal distributions of the two-dimensional points for the Monte Carlo integration
flambda	named list with the prior distribution of the lambda parameter
relTol	relative tolerance in the computation of the kummer function. Default value is 1e-6
nSim	number of two-dimensional points to generate to compute the integral. Default value is 1e4
nStrata	integer vector of length 2 with the number of strata in each dimension. Default value is c(1, 1e2)
verbose	logical (default FALSE) to report progress of the computation

Details

The points are generated according to the accept-reject method using as candidate distribution a Cauchy distribution whose parameters are taken from the prior distributions and the mode of the posterior distribution of the lambda parameter.

The prior distributions are specified as named lists where the first component of each list must be the name of distribution ('unif', 'triang', 'degen', 'gamma') and the rest components must be named according to the name of the parameters of the random generator of the corresponding distribution according to:

- unif: xMin, xMax for the minimum, maximum of the sampled interval.
- degen: x0 for the degenerate value of the random variable.
- triang: xMin, xMax, xMode for minimum, maximum and mode (see [qtriang](#)).
- gamma: scale and shape with the same meaning as in [rgamma](#).

Value

rlambda generates n points according to the posterior distribution of the parameter lambda. The function returns a vector with these points.

See Also

[dlambda](#), [rg](#) for related functions.

Examples

```
# It takes a couple of minutes
hist(rN0(500, nMNO = 20, nReg = 115, fu = list('unif', xMin = 0.3, xMax = 0.5),
        fv = list('unif', xMin = 100, xMax = 120),
        flambda = list('gamma', shape = 11, scale = 12))$N0,
     breaks = seq(1, 200, by = 1), main = '', xlab = 'number of individuals')
```

rmatProb

*Generate matrices of transition probabilities***Description**

Generate a list of matrices of transition probabilities computed with the transition matrices of individuals among pairs of cells detected by the network and specified probability input distributions per cell.

Usage

```
rmatProb(n, nMNOmat, distNames, variation)
```

Arguments

n	number of matrices to generate
nMNOmat	transition matrix with the number of individuals displaced from cell to cell detected by the Mobile Network Operator
distNames	character vector with the names of the prior distributions for each cell
variation	list of lists whose components are parameters providing a measure of variation of each prior distribution

Details

The function generates the probabilities according to a Dirichlet distribution with parameters generated by [alphaPrior](#). These parameters are generated with distributions whose names are taken from the input parameter distNames and construct the corresponding prior distribution for each cell j with mode at $u_j^* = N_j$, where N_j is taken from the sum of rows of nMNOmat. Next the rest of parameters of the distribution are computed according to the dispersion parameters specified in variation.

As accepted distribution names, currently the user can specify unif, degen, triang, and gamma.

The dispersion parameters recognised so far are the coefficients of variation only (standard deviation divided by the mean of the distribution). These dispersion parameters must be specified by a named component cv with a numeric value in $[0, 1]$.

For each distribution the parameters are computed as follows:

- **unif**: This is the uniform distribution with parameters xMax and xMin. Both parameters are computed by $u_j^* \cdot (1 \pm \sqrt{3}cv)$, respectively, in each cell j .
- **degen**: This is the degenerate distribution with parameter x_0 taken as u_j^* in each cell j .
- **triang**: This is the triangular distribution [triang](#) with parameters xMax, xMin, and xMode. The latter is taken directly from nMNOfrom. The distribution is assumed to be symmetrical so that the two former parameters are computed by $u_j^* \cdot (1 \pm \sqrt{3}cv)$, respectively, in each cell j .
- **gamma**: This is the gamma distribution with parameters shape and scale. The former is computed as $\frac{1}{cv^2}$ and the latter as $frac{u_j^*}{scale} - 1$.

Value

A list of n matrices with transition probabilities

Examples

```
nMNOmat <- rbind(c(10, 3, 4), c(5, 21, 3), c(3, 9, 18))
distNames <- rep('unif', 3)
variation <- rep(list(list(cv = 0.20)), 3)
rmatProb(10, nMNOmat, distNames, variation)
```

rN0	<i>Generation of random deviates of the posterior distribution of initial population counts.</i>
-----	--

Description

Generate random points according to the posterior probability distribution of the number of individuals in the hierarchical model.

Usage

```
rN0(n, nMNO, nReg, fu, fv, flambda, scale = 1, relTol = 1e-06,
    nSim = 10000, nStrata = c(1, 100), verbose = FALSE)
```

Arguments

n	number of values to generate
nMNO,	nReg non-negative integer vectors with the number of individuals detected in each cell according to the network operator and the register
fu,	fv named lists with the prior marginal distributions of the two-dimensional points for the Monte Carlo integration
flambda	named list with the prior distribution of the lambda parameter
scale	numeric vector with the scale to count the number of individuals. Default value is 1
relTol	relative tolerance in the computation of the kummer function. Default value is 1e-6
nSim	number of two-dimensional points to generate to compute the integral. Default value is 1e4
nStrata	integer vector of length 2 with the number of strata in each dimension. Default value is c(1, 1e2)
verbose	logical (default FALSE) to report progress of the computation

Details

The posterior distribution is a Poisson distribution with parameter $\lambda * \text{scale}$, where the values of λ are generated with the function [rlambda](#).

The prior distributions are specified as named lists where the first component of each list must be the name of distribution ('unif', 'triang', 'degen', 'gamma') and the rest components must be named according to the name of the parameters of the random generator of the corresponding distribution according to:

- unif: xMin, xMax for the minimum, maximum of the sampled interval.

- `degen`: `x0` for the degenerate value of the random variable.
- `triang`: `xMin`, `xMax`, `xMode` for minimum, maximum and mode (see [qtriang](#)).
- `gamma`: scale and shape with the same meaning as in [rgamma](#).

Value

`rN0` generates `n` points according to the posterior distribution. The function returns a [data.table](#) with these points (under the column `N0`) together with the additional variables:

- The common length of `nMNO` and `nReg` identifies the number of territorial cells in which the number of individuals detected by the telecommunication network and official data. The column `cellID` identifies these territorial cells.
- The different values of the generated values of `lambda` are returned under the column `lambda`.
- The inputs `nMNO` and `nReg` are also included in the output [data.table](#) in columns under the same name.

See Also

[rlambda](#), [rg](#), [rNt](#) for related functions.

Examples

```
# It takes a couple of minutes
hist(rN0(500, nMNO = 20, nReg = 115, fu = list('unif', xMin = 0.3, xMax = 0.5),
      fv = list('unif', xMin = 100, xMax = 120),
      flambda = list('gamma', shape = 11, scale = 12))$N0,
     breaks = seq(1, 200, by = 1), main = '', xlab = 'number of individuals')
```

rNt	<i>Generation of random deviates of the posterior distribution of population counts.</i>
-----	--

Description

Generate random points according to the posterior probability distribution of the number of individuals in the hierarchical model at arbitrary time instants.

Usage

```
rNt(n, nMNOmat, nReg, fu, fv, flambda, distNames, variation, scale = 1,
    relTol = 1e-06, nSim = 1000, nStrata = c(1, 100), verbose = FALSE)
```

Arguments

<code>n</code>	number of values to generate
<code>nMNOmat</code>	transition matrix with the number of individuals displaced from cell to cell detected by the Mobile Network Operator
<code>nReg</code>	non-negative integer vectors with the number of individuals detected in each cell according to the network operator and the register

fu,	fv named lists with the prior marginal distributions of the two-dimensional points for the Monte Carlo integration
flambda	named list with the prior distribution of the lambda parameter
distNames	character vector with the names of the prior distributions for each cell
variation	list of lists whose components are parameters providing a measure of variation of each prior distribution
scale	numeric vector with the scale to count the number of individuals. Default value is 1
relTol	relative tolerance in the computation of the kummer function. Default value is 1e-6
nSim	number of two-dimensional points to generate to compute the integral. Default value is 1e4
nStrata	integer vector of length 2 with the number of strata in each dimension. Default value is c(1, 1e2)
verbose	logical (default FALSE) to report progress of the computation

Details

The posterior distribution is a Poisson distribution with parameter $\text{lambda} * \text{scale}$, where the values of lambda are generated with the function [rlambda](#).

The prior distributions are specified as named lists where the first component of each list must be the name of distribution ('unif', 'triang', 'degen', 'gamma') and the rest of components must be named according to the name of the parameters of the random generator of the corresponding distribution according to:

- unif: xMin, xMax for the minimum, maximum of the sampled interval.
- degen: x0 for the degenerate value of the random variable.
- triang: xMin, xMax, xMode for minimum, maximum and mode (see [qtriang](#)).
- gamma: scale and shape with the same meaning as in [rgamma](#).

Value

rNt generates n points according to the posterior distribution. The function returns a [data.table](#) with these points (under the column N0) together with the additional variables:

- The common length of nMNO and nReg identifies the number of territorial cells in which the number of individuals detected by the telecommunication network and official data. The column cellID identifies these territorial cells.
- The different values of the generated values of lambda are returned under the column lambda.
- The inputs nMNO and nReg are also included in the output [data.table](#) in columns under the same name.

See Also

[rlambda](#), [rg](#), [rNt](#) for related functions.

Examples

```
## First, the inputs:
# The number of generated values
n <- 1e3

#The transition matrix of individuals detected by the MNO
nMNOmat <- rbind(c(10, 3, 4), c(5, 21, 3), c(3, 9, 18))

# Population at the initial time of each cell according to the population register
nReg <- c(90, 130, 101)

# List of priors for u
u0 <- rowSums(nMNOmat) / nReg
cv_u0 <- 0.15
fu <- lapply(u0, function(u){
  umin <- max(0, u - cv_u0 * u)
  umax <- min(1, u + cv_u0 * u)
  output <- list('unif', xMin = umin, xMax = umax)
  return(output)
})

# List of priors for v
v0 <- nReg
cv_v0 <- 0.10
fv <- lapply(v0, function(u){
  umin <- max(0, u - cv_v0 * u)
  umax <- u + cv_v0 * u
  output <- list('unif', xMin = umin, xMax = umax)
  return(output)
})

# List of priors for lambda
cv_lambda <- 0.6
alpha <- 1 / cv_lambda**2 - 1
flambda <- lapply(v0, function(v){list('gamma', shape = 1 + alpha, scale = v / alpha)})

# Names and parameters of priors for the transition probabilities
distNames <- rep('unif', 3)
variation <- rep(list(list(cv = 0.20)), 3)

# The output
Nt <- rNt(n, nMNOmat, nReg, fu, fv, flambda, distNames, variation)$N
hist(Nt, breaks = seq(1, max(Nt) + 10, by = 1), main = '', xlab = 'number of individuals')
```

rNtcondN0

Conditioned generation of random deviates of the posterior distribution of population counts.

Description

Generate random deviates of the posterior distribution of the number of individuals at an arbitrary time instant conditioned upon the initial population.

Usage

```
rNtcondN0(n, N0, nMNOmat, distNames, variation)
```

Arguments

n	number of values to generate
N0	initial population in each cell
nMNOmat	transition matrix with the number of individuals displaced from cell to cell detected by the Mobile Network Operator
distNames	character vector with the names of the prior distributions for each cell
variation	list of lists whose components are parameters providing a measure of variation of each prior distribution

Details

The function generates the probabilities according to a Dirichlet distribution with parameters generated by [alphaPrior](#). These parameters are generated with distributions whose names are taken from the input parameter `distNames` and construct the corresponding prior distribution for each cell j with mode at $u_j^* = N_j$, where N_j is taken from the sum of rows of `nMNOmat`. Next the rest of parameters of the distribution are computed according to the dispersion parameters specified in `variation`.

As accepted distribution names, currently the user can specify `unif`, `degen`, `triang`, and `gamma`.

The dispersion parameters recognised so far are the coefficients of variation only (standard deviation divided by the mean of the distribution). These dispersion parameters must be specified by a named component `cv` with a numeric value in $[0, 1]$.

For each distribution the parameters are computed as follows:

- `unif`: This is the uniform distribution with parameters `xMax` and `xMin`. Both parameters are computed by $u_j^* \cdot (1 \pm \sqrt{3}cv)$, respectively, in each cell j .
- `degen`: This is the degenerate distribution with parameter `X0` taken as u_j^* in each cell j .
- `triang`: This is the triangular distribution [triang](#) with parameters `xMax`, `xMin`, and `xMode`. The latter is taken directly from `nMNOfrom`. The distribution is assumed to be symmetrical so that the two former parameters are computed by $u_j^* \cdot (1 \pm \sqrt{3}cv)$, respectively, in each cell j .
- `gamma`: This is the gamma distribution with parameters `shape` and `scale`. The former is computed as $\frac{1}{cv^2}$ and the latter as $frac{u_j^*}{scale} - 1$.

Value

Return a matrix with as many columns as cells and `n` rows with the generated values

Examples

```
N0 <- c(93, 123, 130)
nMNOmat <- rbind(c(10, 3, 4), c(5, 21, 3), c(3, 9, 18))
distNames <- rep('unif', 3)
variation <- rep(list(list(cv = 0.20)), 3)
rNtcondN0(1e3, N0, nMNOmat, distNames, variation)
```

rp	<i>Generate random vector deviates of transition probabilities.</i>
----	---

Description

Generate random vector deviates of the transition probabilities $p_{ij}(t_0, t_n)$ for a given cell i stacked into an $n \times (\text{number of cells})$ matrix

Usage

```
rp(n, flist)
```

Arguments

n	number of probability vectors to generate
flist	list with the prior distributions for each cell

Details

The prior distributions are specified as named lists where the first component of each list must be the name of distribution ('unif', 'triang', 'degen', 'gamma') and the rest components must be named according to the name of the parameters of the random generator of the corresponding distribution according to:

- unif: xMin, xMax for the minimum, maximum of the sampled interval.
- degen: x0 for the degenerate value of the random variable.
- triang: xMin, xMax, xMode for minimum, maximum and mode (see [qtriang](#)).
- gamma: scale and shape with the same meaning as in [rgamma](#).

Value

Return a matrix with n rows and as many columns as cells taken from the length of flist. Each row is thus a probability vector

Examples

```
flist <- alphaPrior(c(10, 3, 4), c('unif', 'triang', 'gamma'),
  list(list(cv = 0.1), list(cv = 0.05), list(cv = 0.15)))
rp(10, flist)
```

triang	<i>The Triangular Distribution.</i>
--------	-------------------------------------

Description

Density, distribution function, quantile function and random generation for the triangular distribution

Usage

```
dtriang(x, xMin, xMax, xMode)
```

```
ptriang(q, xMin, xMax, xMode)
```

```
qtriang(q, xMin, xMax, xMode)
```

```
rtriang(n, xMin, xMax, xMode)
```

Arguments

x,	q vector of quantiles
xMin	vector with the minimum values of the range of the random variable
xMax	vector with the maximum values of the range of the random variable
xMode	vector with the modes of the random variable
n	number of observations
p	vector of probabilities

Value

dtriang gives the density, ptriang gives the distribution function, qtriang gives the quantile function, and rtriang generates random deviates.

The lengths of the input vectors (except n) must be all equal except when their length is 1. Otherwise NAs are produced.

See Also

[Distributions](#) for other distributions

Examples

```
curve(dtriang(x, 0, 3, 1), xlim = c(0, 3))
curve(ptriang(x, 0, 3, 1), xlim = c(0, 3))
curve(qtriang(x, 0, 3, 1), xlim = c(0, 1))
hist(rtriang(1e6, 0, 3, 1), breaks = seq(0, 3, by = 0.01))
```

Index

*Topic **datasets**

- flambda, 7
- fu, 7
- fv, 8
- MobPop, 11
- nMNO_ini, 13
- nReg, 13

alphaPrior, 3, 15, 24, 29

beta, 21

data.table, 6–11, 26, 27

dg, 4

Distributions, 31

dlambda, 5, 5, 12, 15, 22, 23

dpois, 5

dtriang, 3, 15

dtriang(triang), 31

flambda, 7

fu, 7

fv, 8

genAlpha, 8, 15

genUV, 6–8, 9, 15

kummer, 4, 5, 10, 12, 14, 15, 17, 18, 22, 23, 25, 27

lbeta, 21

MobPop, 11

modeLambda, 5, 11, 14, 22

nMNO_ini, 13

nReg, 13

pestim, 13

pestim-package (pestim), 13

Phi, 6, 14, 15

pkummer, 16

postN0, 15, 16, 19

postNt, 15, 18

postNtcondN0, 15, 19, 20

ptriang, 15

ptriang(triang), 31

qtriang, 6, 8–10, 12, 15, 17, 18, 23, 26, 27, 30

qtriang(triang), 31

ratioBeta, 14, 15, 21

rg, 15, 21, 23, 26, 27

rgamma, 6, 8–10, 12, 17, 18, 23, 26, 27, 30

rlambda, 15, 22, 25–27

rmatProb, 15, 24

rN0, 15, 17, 25

rNt, 15, 19, 26, 26, 27

rNtcondN0, 15, 28

rp, 15, 30

rtriang, 15

rtriang(triang), 31

runif, 10

triang, 24, 29, 31