Package 'inference'

October 28, 2020

Type Package

Title R package for computing the number of individuals in the target population conditioned on the number of individuals detected by the MNO and auxiliary information.

Version 0.1.0

Author Bogdan Oancea <bogdan.oancea@gmail.com>, David Salgado <david.salgado.fernandez@ine.es> Sandra Barragan <sandra.barragan.andres@ine.es>

Maintainer Bogdan Oancea <bogdan.oancea@gmail.com>

Description R package for computing the number of individuals in the target population conditioned on the number of individuals detected by the mobile network and some auxiliary information

License GPL3, EUPL

Imports data.table,
deduplication,
Matrix,
doParallel,
parallel,
extraDistr

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Suggests knitr,
rmarkdown

VignetteBuilder knitr

Collate 'buildCluster.R'
'computeDeduplicationFactors.R'
'computeDistrParams.R'
'utils.R'
'computeInitialPopulation.R'
'computeTau.R'
'computeStats.R'
'computePopulationOD.R'
'computePopulationT.R'
'computeRegionAreas.R'
'example.R'
'exampleAPI.R'

'inference.R'
'readNnetInitial.R'

R topics documented:

`computeDeduplicationFactors` *Computes the deduplication factors for each region.*

Description

Computes the deduplication factors for each region of the map. For a complete description of these factors an interested reader can consult the description of the methodological framework https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/images/f/fb/WPI_Deliverable_I3_A_proposed_production_framework_with_mobile_network_data_2020_05_31_draft.pdf.

Usage

```
computeDeduplicationFactors(
  dupFileName,
  regsFileName,
  postLocPrefix,
  postLocPath
)
```

Arguments

`dupFileName`	The name of the file with the duplicity probabilities. This file is the output of the `deduplication` package.
`regsFileName`	The name of the .csv file defining the regions. It has two columns: `tile,region`. The first column contains the IDs of each tile in the grid while the second contains the number of a region. This file is defined by the user and it can be created with any text editor.
`postLocPrefix`	The file name prefix of the files with posterior location probabilities for each device. The whole file name is composed by a concatenation of `prefixName`, `_` and `deviceID`. The extension of these files is `.dt.csv`
`postLocPath`	The path to the location where the posterior location probabilities are stored.

Value

A data.table object with the deduplication factors for each region.

References

https://github.com/MobilePhoneESSnetBigData

`computeDistrParams` *Computes the parameters of the population counts distributions.*

Description

Computes a series of parameters needed to build the target population counts distribution. For a complete description of these parameters an interested reader can consult the description of the methodological framework https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/images/f/fb/WPI_Deliverable_I3_A_proposed_production_framework_with_mobile_network_data_2020_05_31_draft.pdf.

Usage

```
computeDistrParams(
  omega,
  popRegFileName,
  pntRateFileName,
  regsFileName = NULL,
  gridFileName = NULL,
  rel_bias = 0,
  cv = 1e-05
)
```

Arguments

`omega`	
`popRegFileName`	The name of the file with the population counts for each region taken from a population register. It has 2 columns: `region,N0`.
`pntRateFileName`	The name of the file with the penetration rates for each region. It has 2 columns: `region,pntRate`.
`regsFileName`	The name of the .csv file defining the regions. It has two columns: `tile,region`. The first column contains the IDs of each tile in the grid while the second contains the number of a region. This file is defined by the user and it can be created with any text editor. It is required only for the state process negative binomial distribution.
`gridFileName`	The name of the .csv file with the grid parameters. It is required only for the state process negative binomial distribution.
`cv`	The coefficient of variation for the population density of each region. The default value is 0.
`re_bias`	The value of the relative bias for the population density of each region. The default value is 0.

Value

A data.table object with the following columns region,omega1,omega2,pnrRate,regionArea_km2,N0,dedupPntRate
If regsFileName and gridFileName are not NULL the result will have 3 more columns:region,omega1,omega2,pnrRa
They are needed only for the state process negative binomial distribution.

References

https://github.com/MobilePhoneESSnetBigData

`computeInitialPopulation` *Computes the distribution of the population count at initial time instant.*

Description

Computes the distribution of the population count at initial time instant using one of the three distributions: Negative Binomial, Beta Negative Binomial or State Process Negative Binomial. For details of the theoretical background behind this distribution an interested reader can consult the description of the methodological framework https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/images/f/fb/WPI_Deliverable_I3_A_proposed_production_framework_with_mobile_network_data_2020_05_31_draft.pdf.

Usage

```
computeInitialPopulation(
  nnet,
  params,
  popDistr,
  rndVal = FALSE,
  ciprob = NULL,
  method = "ETI"
)
```

Arguments

nnet	The random values generated with aggregation package for the number of individuals detected by the network.
params	The parameters of the distribution. It should be a data.table object with the following columns: region,omega1,omega2,pnrRate,regionArea_km2,N0,dedupPntRate,alpha,b
popDistr	The distribution to be used for population count. This parameter could have one of the following values: NegBin (negative binomial distribution), BetaNegBin (beta negative binomial distribution) or STNegBin (state process negative binomial distribution).
rndVal	If FALSE the result return by this function will be a list with a single element, a data.table object with the following columns: region,Mean,Mode,Median,SD,Min,Max,Q1,Q3,IQR If TRUE the list will have a second element which is a data.table object containing the random values generated for each region.
ciprob	Value of probability of the CI (between 0 and 1) to be estimated. If NULL the default value is 0.89.

method	The method to compute credible intervals. It could have 2 values, 'ETI' or 'HDI'. The default value is 'ETI.

Value

A list object with one or two elements. If rndVal is FALSE the list will have a single element with descriptive statistics for the population count, which is a data.table object with the following columns: region,Mean,Mode,Median,Min,Max,Q1,Q3,IQR,SD,CV,CI_LOW,CI_HIGH. If rndVal is TRUE the list will have a second element which is a data.table object containing the random values generated for each region. The name of the two list elements giving the descriptive statistics and random values for time t are 'stats' and 'rnd_values'.

References

https://github.com/MobilePhoneESSnetBigData

---

computePopulationOD *Computes the origin-destination matrices.*

Description

Computes the origin-destination matrices for all pairs of time instants time_from-time_to. For details of the theoretical background of the origin-destination matrices computation an interested reader can consult the description of the methodological framework https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/images/f/fb/WPI_Deliverable_I3_A_proposed_production_framework_with_mobile_network_data_2020_05_31_draft.pdf.

Usage

```
computePopulationOD(
  nt0,
  nnetODFileName,
  zip = TRUE,
  rndVal = FALSE,
  ciprob = NULL,
  method = "ETI"
)
```

Arguments

nt0	The population at t0.
nnetODFileName	the name of the file where the population moving from one region to another is stored. This is an output of the aggregation package.
zip	If TRUE the file where where the population moving from one region to another is stored is a zipped csv file, otherwise it is simple csv file.
rndVal	Controls if the random values generated for each t are returned or not in the result of this function. If TRUE, the random values generated according to the corresponding distribution are returned in the results, if FALSE only the summary statistics for each t and region are returned.
ciprob	Value of probability of the CI (between 0 and 1) to be estimated. If NULL the default value is 0.89.

`method` The method to compute credible intervals. It could have 2 values, 'ETI' or 'HDI'. The default value is 'ETI.

Value

A list with one element for each pair of time_from-time_to. Each element of the list is also a list with one or two elements, depending on the value of the rndVal parameter. If rndVal is TRUE there are two elements in the list corresponding to time instant a pair time_from-time_to. The first one is a data.table object with some descriptive statistics for the origin-destination matrix, containing the following columns:`region_from`,`region_to`,`Mean`,`Mode`,`Median`,`Min`,`Max`,`Q1`,`Q3`,`IQR`,`SD`,`CV`,`CI_LOW`,`CI_HIGH`. The second one is a data.table object with the random values for origin-destination matrix generated for each pair of time instants time_from-time_to and each pair of regions region_from-region_to, with the following columns: `region_from`,`region_to`,`iter`,`NPop`. If rndVal is FALSE the list for a pair of time instants time-from-time_to contains only the first element previously mentioned. The name of the list element corresponding to a pair of time instants is 'time_from-time_to' and the name of the two list elements giving the descriptive statistics and random values are 'stats' and 'rnd_values'.

References

https://github.com/MobilePhoneESSnetBigData

---

computePopulationT *Computes population counts at time instants t >t0.*

Description

Computes the distribution of the population counts for all times instants t > t0. For details of the theoretical background behind this distribution an interested reader can consult the description of the methodological framework https://webgate.ec.europa.eu/fpfis/mwikis/essnetbigdata/images/f/fb/WPI_Deliverable_I3_A_proposed_production_framework_with_mobile_network_data_2020_05_31_draft.pdf.

Usage

```
computePopulationT(
  nt0,
  nnetODFileName,
  zip = TRUE,
  rndVal = FALSE,
  ciprob = NULL,
  method = "ETI"
)
```

Arguments

`nt0`	The population at t0.
`nnetODFileName`	the name of the file where the population moving from one region to another is stored. This is an output of the `aggregation` package.
`zip`	If TRUE the file where where the population moving from one region to another is stored is a zipped csv file, otherwise it is simple csv file.

rndVal	Controls if the random values generated for each t >t0 are returned or not in the result of this function. If TRUE, the random values generated according to the corresponding distribution are returned in the results, if FALSE only the summary statistics for each t>t0 and region are returned.
ciprob	Value of probability of the CI (between 0 and 1) to be estimated. If NULL the default value is 0.89.
method	The method to compute credible intervals. It could have 2 values, 'ETI' or 'HDI'. The default value is 'ETI.

Value

A list with one element for each time instant (including t0). Each element of the list is also a list with one or two elements, depending on the value of the rndVal parameter. If rndVal is TRUE there are two elements in the list corresponding to time instant t. The first one is a data.table object with some descriptive statistics for the population count at time t, containing the following columns:`region,Mean,Mode,Median,Min,Max,Q1,Q3,IQR,SD,CV,CI_LOW,CI_HIGH`. The second one is a data.table object with the random values for population count generated for each region, with the following columns: `region,iter,NPop`. If rndVal is FALSE the list for time instant t contains only the first element previously mentioned. The name of the list element corresponding to time instant t is 't' and the name of the two list elements giving the descriptive statistics and random values for time t are 'stats' and 'rnd_values'.

References

https://github.com/MobilePhoneESSnetBigData

---

example *Example of using the inference package*

Description

This is just an example on how to use this package to generate the distribution of the population count.

Usage

```
example()
```

Details

This is a script that shows how to use the functions of this package to compute the distribution of the initial target population count, the distribution of the population count at successive time instants and the origin-destination matrix.

References

https://github.com/MobilePhoneESSnetBigData

Examples

```
# set the folder where the necessary input files are stored and the prefix of the input file names.
path      <- 'extData'

prefix <- 'postLocDevice'

# compute the deduplication factors
dpFileName <- system.file(path, 'duplicity.csv', package = 'inference')
rgFileName <- system.file(path, 'regions.csv', package = 'inference')

omega_r <- computeDeduplicationFactors(dpFileName, rgFileName, prefix, system.file(path, package = 'inference

# reads the number of individuals detected by network
nFileName <- system.file(path, 'nnet.csv', package = 'inference')
nnet <- readNnetInitial(nFileName)

# compute the parameters of the distribution
pRFileName <- system.file(path, 'pop_reg.csv', package = 'inference')
pRateFileName <- system.file(path, 'pnt_rate.csv', package = 'inference')
grFileName <- system.file(path, 'grid.csv', package = 'inference')
params <- computeDistrParams(omega_r, pRFileName, pRateFileName, rgFileName, grFileName)

# A. Compute the population count distribution at t0
# compute the population count distribution using the Beta Negative Binomial distribution
n_bnb <- computeInitialPopulation(nnet, params, popDistr = 'BetaNegBin', rndVal = TRUE)

# display results
n_bnb$stats
head(n_bnb$rnd_values)


# compute the population count distribution using the Negative Binomial distribution
n_nb <- computeInitialPopulation(nnet, params, popDistr = 'NegBin', rndVal = TRUE)

# display results
n_nb$stats
head(n_nb$rnd_values)


# compute the population count distribution using the state process Negative Binomial distribution
n_stnb <- computeInitialPopulation(nnet, params, popDistr= 'STNegBin', rndVal = TRUE)

# display results
n_stnb$stats
head(n_stnb$rnd_values)


# B. compute the population count distribution at time instants t > t0
# first set the name of the file with the population moving from one region to another (output of the aggregation
nnetODFile <- system.file(path, 'nnetOD.zip', package = 'inference')

# 1.Using the Beta Negative Binomial distribution
nt_bnb <- computePopulationT(n_bnb$rnd_values, nnetODFile, rndVal = TRUE)
```

```
# display results
# first, select a random time instant
times <- names(nt_bnb)
t <- sample(1:length(times), size = 1)
t
nt_bnb[[t]]$stats
head(nt_bnb[[t]]$rnd_values)


# 2.Using the Negative Binomial distribution
nt_nb <- computePopulationT(n_nb$rnd_values, nnetODFile, rndVal = TRUE)

# display results
# first, select a random time instant
times <- names(nt_nb)
t <- sample(1:length(times), size = 1)
t
nt_nb[[t]]$stats
head(nt_nb[[t]]$rnd_values)
# 3.Using the state process Negative Binomial distribution
nt_stnb <- computePopulationT(n_stnb$rnd_values, nnetODFile, rndVal = TRUE)

# display results
# first, select a random time instant
times <- names(nt_stnb)
t <- sample(1:length(times), size = 1)
t
nt_stnb[[t]]$stats
head(nt_stnb[[t]]$rnd_values)

# C. compute the origin-destination matrices for all pairs of time instants time_from-time_to
# first set the name of the file with the population moving from one region to another (output of the aggregation
nnetODFile <- system.file(path, 'nnetOD.zip', package = 'inference')

# 1.Using the Beta Negative Binomial distribution
OD_bnb <- computePopulationOD(n_bnb$rnd_values, nnetODFile, rndVal = TRUE)

# display results
time_pairs <- names(OD_bnb)
# first, select a random time instants pair
i <- sample(1:length(time_pairs), size = 1)
time_pairs[i]
OD_bnb[[i]]$stats
head(OD_bnb[[i]]$rnd_values)

# 2.Using the Negative Binomial distribution
OD_nb <- computePopulationOD(n_nb$rnd_values, nnetODFile, rndVal = TRUE)

# display results
time_pairs <- names(OD_nb)
# first, select a random time instants pair
i <- sample(1:length(time_pairs), size = 1)
time_pairs[i]
OD_nb[[i]]$stats
head(OD_nb[[i]]$rnd_values)

# 3.Using the  state process Negative Binomial distribution
```

```
OD_stnb <- computePopulationOD(n_stnb$rnd_values, nnetODFile, rndVal = TRUE)

# display results
time_pairs <- names(OD_stnb)
# first, select a random time instants pair
i <- sample(1:length(time_pairs), size = 1)
time_pairs[i]
OD_stnb[[i]]$stats
head(OD_stnb[[i]]$rnd_values)
```

---

exampleAPI *Example of using the API inference*

Description

This is just an example on how to use the REST API of this package to generate the distribution of the population count.

Usage

```
exampleAPI()
```

Details

This is a script that shows how to use the REST API of this package to compute the distribution of the initial target population count, the disitribution of the population count at successive time instants and the origin-destination matrix.

References

https://github.com/MobilePhoneESSnetBigData

Examples

```
########################################
# First, in a separate R console run the following instructions that start the http API on your local computer
# One can replace 127.0.0.1 with the API address of another Web server
library(plumber)
pathPL <-'plumber'
initPop_api <- plumber::plumb(system.file(pathPL, 'plumb.R', package = 'inference'))
initPop_api$run(host = '127.0.0.1', port = 8000)
########################################

library(httr)
library(jsonlite)
library(data.table)

# set the folder where the necessary input files are stored and the prefix of the input file names.
path      <- 'extData'
prefix <- 'postLocDevice'
postLocPath <- system.file(path, package = 'inference')
```

```
# compute the deduplication factors
dpFileName <- system.file(path, 'duplicity.csv', package = 'inference')
rgFileName <- system.file(path, 'regions.csv', package = 'inference')

# prepare the body of the http request
body <- list(
  .dupFileName = dpFileName,
  .regsFileName = rgFileName,
  .postLocPrefix = prefix,
  .postLocPath = postLocPath
)

# set API path
pathDedup <- 'computeDeduplicationFactors'

# send POST Request to API
url <- "http://127.0.0.1:8000"
raw.result <- POST(url = url, path = pathDedup, body = body, encode = 'json')

# check status code
raw.result$status_code

# transform back the results from json format
omega_r <- as.data.table(fromJSON(rawToChar(raw.result$content)))

# Compute the parameters of the distribution
# First reads the number of individuals detected by network
nFileName <- system.file(path, 'nnet.csv', package = 'inference')
nnet <- readNnetInitial(nFileName)

pRFileName <- system.file(path, 'pop_reg.csv', package = 'inference')
pRateFileName <- system.file(path, 'pnt_rate.csv', package = 'inference')
grFileName <- system.file(path, 'grid.csv', package = 'inference')

# prepare the body of the http request
body <- list(
   .omega = omega_r,
  .popRegFileName = pRFileName,
  .pntRateFileName  = pRateFileName,
  .regsFileName = rgFileName,
  .gridFileName = grFileName,
  .rel_bias = 0,
  .cv = 1e-5
)

# set API path
pathDistr <- 'computeDistrParams'

 # send POST Request to API
raw.result <- POST(url = url, path = pathDistr, body = body, encode = 'json')

# check status code
raw.result$status_code

# transform back the results from json format
params <- as.data.table(fromJSON(rawToChar(raw.result$content)))
```

```
# Compute the population count distribution at t0 using the Beta Negative Binomial distribution

# prepare the body of the http request
body <- list(
  .nnet = nnet,
  .params = params,
  .popDistr = 'BetaNegBin',
  .rndVal = TRUE,
  .ciprob = 0.95,
  .method = 'ETI'
)

# set API path
pathInit <- 'computeInitialPopulation'

 # send POST Request to API
raw.result <- POST(url = url, path = pathInit, body = body, encode = 'json')

# check status code
raw.result$status_code

# transform back the results from json format
n_bnb <- fromJSON(rawToChar(raw.result$content))

# display results
n_bnb$stats
head(n_bnb$rnd_values)

# Compute the population count distribution at time instants t > t0 using the Beta Negative Binomial distributio
# first set the name of the file with the population moving from one region to another (output of the aggregation
nnetODFile <- system.file(path, 'nnetOD.zip', package = 'inference')

# prepare the body of the http request
body <- list(
  .nt0 = as.data.table(n_bnb$rnd_values),
  .nnetODFileName = nnetODFile,
  .zip = TRUE,
  .rndVal = TRUE,
  .ciprob = 0.95,
  .method = 'ETI'
)

# set API path
pathT <- 'computePopulationT'

 # send POST Request to API
raw.result <- POST(url = url, path = pathT, body = body, encode = 'json')

# check status code
raw.result$status_code

# transform back the results from json format
nt_bnb <- fromJSON(rawToChar(raw.result$content))

# display results
# first, select a random time instant
times <- names(nt_bnb)
```

```
t <- sample(1:length(times), size = 1)
t
nt_bnb[[t]]$stats
head(nt_bnb[[t]]$rnd_values)


# Compute the Origin-Destination matrices for all pairs of time instants time_from-time_to using the Beta Negati

# prepare the body of the http request
body <- list(
  .nt0 = as.data.table(n_bnb$rnd_values),
  .nnetODFileName = nnetODFile,
  .zip = TRUE,
  .rndVal = TRUE,
  .ciprob = 0.95,
  .method = 'ETI'
)

# set API path
pathOD <- 'computePopulationOD'

 # send POST Request to API
raw.result <- POST(url = url, path = pathOD, body = body, encode = 'json')

# check status code
raw.result$status_code

# transform back the results from json format
OD_bnb <- fromJSON(rawToChar(raw.result$content))

# display results
time_pairs <- names(OD_bnb)
# first, select a random time instants pair
i <- sample(1:length(time_pairs), size = 1)
time_pairs[i]
OD_bnb[[i]]$stats
head(OD_bnb[[i]]$rnd_values)
```

`inference` *inference: A package for computing the distribution of the number of individuals in the target population conditioned on the number of individuals detected by MNO.*

Description

This package contains functions to compute the distribution of the number of individuals in the target population conditioned on the number of individuals detected by MNO. For an example on how to use this package please read example.R or exampleAPI.R.

`readNnetInitial` *Reads the number of individuals detected by network at initial time.*

Description

Reads the number of individuals detected by network at initial time instant, as they are generated by the `aggregation` package.

Usage

```
readNnetInitial(nnetFileName)
```

Arguments

`nnetFileName`	The file name of the random values generated by the `aggregation` package for the number of individuals detected by network for each region and each time instant.

Value

A data.table object with two columns: `region,N`

Index