

# Package ‘jollofR’

June 11, 2025

**Type** Package

**Title** A Bayesian statistical model-based approach for disaggregating small area population estimates by demographic characteristics

**Version** 0.3.0

**Description** Automatic dis-aggregation of small area population estimates by demographic groups (e.g., age, sex, race, marital status, educational level, etc) along with the estimates of uncertainty, using advanced Bayesian statistical modelling approaches.

**License** MIT + file LICENSE

**URL** <https://github.com/wpgp/jollof>, <https://wpgp.github.io/jollof/>, <https://wpgp.github.io/jollofR/>

**BugReports** <https://github.com/wpgp/jollof/issues>

**SystemRequirements** R ( $\geq 4.1.0$ )

**Depends** R ( $\geq 4.1.0$ )

**Imports** dplyr,  
ggplot2,  
ggpubr,  
caret,  
reshape2,  
terra,  
rasterVis,  
R6,  
sf,  
utils

**Suggests** INLA,  
haven,  
tibble,  
tidyr,  
glue,  
knitr,  
matrixStats,  
purrr,  
RcppEigen,  
rlang,  
rmarkdown,  
rstudioapi,

```
scales,
sp,
stringr,
testthat (>= 3.0.0)
VignetteBuilder knitr
Config/testthat/edition 3
Encoding UTF-8
LazyData true
Additional_repositories https://inla.r-inla-download.org/R/stable/
Roxygen list(markdown = TRUE, r6 = TRUE)
RoxygenNote 7.3.2
Config/Needs/website rmarkdown
```

Contents

boxLine . . . . .	2
cheesecake . . . . .	3
cheesepop . . . . .	4
plotHist . . . . .	5
plotRast . . . . .	5
pyramid . . . . .	6
slices . . . . .	7
spices . . . . .	8
splash . . . . .	8
splash1 . . . . .	9
spray . . . . .	10
spray1 . . . . .	11
sprinkle . . . . .	12
sprinkle1 . . . . .	13
toydata . . . . .	14
<b>Index</b>	<b>17</b>

---

boxLine	<i>boxLine: Produces two graphs - boxplots of disaggregated population counts across groups and a line plot showing the mean distribution of the disaggregated counts.</i>
---------	--

---

Description

This function automarically generate two graphs that are combined together - a boxplot graph (a) of the distribution of of the various group’s disaggregated population count, and (b) a line graph of the aggregated counts across all groups (e.g., total number of individuals for each group). Its input data could come from any of the disaggregation functions within the 'jollofR' package such as 'cheesecake', 'cheesepop', 'slices', etc.

Usage

```
boxLine(dmat, xlab, ylab)
```

**Arguments**

<code>dmat</code>	A data frame containing the group-structured disaggregated population estimates which could be an observed data or model estimates from 'cheesecake', 'cheesepop', 'slices', 'spices', 'spray', 'sprinkle', 'splash', 'spray', 'sprinkle1', 'splash1', or 'spray1'. considered.
<code>xlab</code>	A user-defined label for the x-axis (e.g., 'Frequency').
<code>ylab</code>	A user-defined label for the y-axis (e.g., 'Population count').

**Value**

A graphic image of combined boxplot and line plots of population counts across groups.

**Examples**

```
data(toydata)
result <- cheesecake(df = toydata$admin, output_dir = tempdir())
boxLine(dmat=result$male_age_pop,
xlab="Age group (years)",
ylab = "Population Count")
```

---

<code>cheesecake</code>	<i>cheesecake: Population disaggregation by two-level demographic groups (eg., age and sex), with covariates</i>
-------------------------	--

---

**Description**

Used to disaggregate small area population estimates by age, sex, and other socio-demographic and socio-economic characteristics (e.g., ethnicity, religion, educational level, immigration status, etc).

It uses statistical models to predict population proportions and population totals across demographic groups. Primarily designed to help users in filling population data gaps across demographic groups due to outdated or incomplete census data.

**Usage**

```
cheesecake(df, output_dir)
```

**Arguments**

<code>df</code>	A data frame object containing sample data (often partially observed) on age and sex groups population data, for example, as well as the overall total population counts per administrative unit.
<code>output_dir</code>	This is the directory with the name of the output folder where the disaggregated population proportions and population totals are automatically saved.

**Value**

A list of data frame objects of the output files including the disaggregated population proportions and population totals along with the corresponding measures of uncertainties (lower and upper bounds of 95-percent credible intervals) for each demographic characteristic. In addition, a file containing the model performance/model fit evaluation metrics is also produced.

**Examples**

```
data(toydata)
result <- cheesecake(df = toydata$admin, output_dir = tempdir())
```

---

cheesepop	<i>cheesepop: Population disaggregation by two-level demographic groups (eg., age and sex), without covariates</i>
-----------	--

---

**Description**

Similar to the 'cheesecake' function, 'cheesepop' disaggregates small area population estimates by age, sex, and other socio-demographic and socio-economic characteristics (e.g., ethnicity, religion, educational level, immigration status, etc). However, unlike the 'cheesecake' function which uses geospatial covariates to predict missing data values, 'cheesepop' does not use covariates.

It uses Bayesian statistical models to predict population proportions and population totals across demographic groups. Primarily designed to help users in filling population data gaps across demographic groups due to outdated or incomplete census data.

**Usage**

```
cheesepop(df, output_dir)
```

**Arguments**

df	A data frame object containing sample data (often partially observed) on age and sex groups population data as well as the estimated overall total counts per administrative unit.
output_dir	This is the directory with the name of the output folder where the disaggregated population proportions and population totals are automatically saved.

**Value**

A list of data frame objects of the output files including the disaggregated population proportions and population totals along with the corresponding measures of uncertainties (lower and upper bounds of 95-percent credible intervals) for each demographic characteristic. In addition, a file containing the model performance/model fit evaluation metrics is also produced.

**Examples**

```
data(toydata)
result <- cheesepop(df = toydata$admin, output_dir = tempdir())
```

---

plotHist	<i>plotHist: Produces histogram of the disaggregated population counts for each group</i>
----------	---

---

### Description

This function produces multi-panel histogram plots of the disaggregated population counts for each group. Its input data can come from any of the disaggregation functions within the 'jollofR' package such as 'cheesecake', 'cheesepop', 'slices', etc.

### Usage

```
plotHist(dmat, xlab, ylab)
```

### Arguments

dmat	A data frame containing the group-structured disaggregated population estimates which could either be observed or predicted from 'cheesecake', 'cheesepop', 'slices', 'spices', 'spray', 'sprinkle', 'splash', 'spray', 'sprinkle1', 'splash1', and 'spray1'.
xlab	A user-defined label for the x-axis (e.g., 'Frequency') considered.
ylab	A user-defined label for the y-axis (e.g., 'Population count') considered.

### Value

A graphic image of histogram of the disaggregated population count

### Examples

```
data(toydata)
result <- cheesecake(df = toydata$admin, output_dir = tempdir())
plotHist(dmat=result$age_pop, xlab="Population Count", ylab = "Frequency")
```

---

plotRast	<i>plotRast: Produces multi-panel maps of the raster files of the grid-cell disaggregated structured population counts</i>
----------	--

---

### Description

This function produces multi-frame maps of the raster files across the various demographic groups of interest. The input data could come from any of the jollofR disaggregation functions at grid cell levels, e.g., 'sprinkle', 'spray', 'splash', 'sprinkle1', 'spray1', 'and splash1',

### Usage

```
plotRast(output_dir, names, nrow, ncol)
```

Arguments

output_dir	The raster files folder directory for the disaggregated population estimates
names	A user-defined names for the x-axis ticks labels.
nrow	Number of rows of the multi-panel maps. The value depends on the number of groups being displayed.
ncol	Number of columns of the multi-panel maps. The value depends on the number of groups being displayed.

Value

A graphic image of multi-panel maps of population disaggregated raster files

Examples

```
dontrun{data(toydata)
# run the appropriate function
result <- cheesepop(df = toydata$admin, output_dir) # run cheesepop or cheesecake
rclass <- paste0("TOY_population_v1_0_age",1:12) # specify the names of the raster files to be saved
class <- names(result$full_data %>% dplyr::select(starts_with("age_"))) # specify the raster files to visualis
output_dir <- paste0(out_path, "/raster/spray2") # set output directory
result2b <- spray(df=result$full_data, rdf=toydata$grid,
                  rclass, output_dir)
#make raster maps
#- use below to see the list of raster files saved in the directory
list.files(output_dir, pattern = "\\\\.tif$", full.names = TRUE)

group <- 1:12 # customised group #
raster_files <- paste0(tempdir(), "/pop_",paste0("TOY_population_v1_0_age",group), ".tif")
names <- paste0("Age ", group) # Customised names of the plot panels (same length as rclass)
nrow = 4; ncol = 3 # rows and columns of the panels of the output maps
plotRast(tempdir(), names, nrow, ncol)} # make the maps
```

---

pyramid	<i>pyramid: Produces population pyramid (graphs) of demographics (for cheesecake and cheesepop age-sex output data)</i>
---------	---

---

Description

This function creates population pyramid for age and sex output data from the 'cheesecake' or 'cheesepop' functions outputs. It could also be used to visualize observed age-sex compositions.

Usage

```
pyramid(female_pop, male_pop)
```

Arguments

female_pop	A data frame containing the disaggregated population estimates for females across all ages groups.
male_pop	A data frame containing the disaggregated population estimates for males across all ages groups.

**Value**

A graphic image of age-sex population distribution pyramid

**Examples**

```
data(toydata)
result <- cheesecake(df = toydata$admin, output_dir = tempdir())
pyramid(result$fem_age_pop, result$male_age_pop)
```

---

slices	<i>slices: Disaggregating population counts for a single level of demographics (e.g., age groups only or sex group only) - without covariates. Please use 'spices' if you want covariates included.</i>
--------	---

---

**Description**

This function disaggregates population estimates by a single demographic (age or sex or religion, etc)

**Usage**

```
slices(df, output_dir, class)
```

**Arguments**

df	A data frame object containing sample data (often partially observed) on age or sex groups population data as well as the estimated overall total counts per administrative unit.
output_dir	This is the directory with the name of the output folder where the disaggregated population proportions and population totals are automatically saved.
class	These are the categories of the variables of interest. For example, for educational level, it could be 'no education', 'primary education', 'secondary education', 'tertiary education'.

**Value**

A list of data frame objects of the output files including the disaggregated population proportions and population totals along with the corresponding measures of uncertainties (lower and upper bounds of 95-percent credible intervals) for each demographic characteristic. In addition, a file containing the model performance/model fit evaluation metrics is also produced.

**Examples**

```
## Not run: data(toydata)
classes <- names(toydata$admin %>% dplyr::select(starts_with("age_")))
result2 <- slices(df = toydata$admin, output_dir = tempdir(), class = classes)
## End(Not run)
```

---

spices	<i>spices: Disaggregates population counts for a single level of demographics (e.g., age groups only or sex group only) with covariates.</i>
--------	--

---

### Description

This function disaggregates population estimates by a single demographic (age or sex or religion, etc)

### Usage

```
spices(df, output_dir, class)
```

### Arguments

df	A data frame object containing sample data (often partially observed) on age or sex groups population data as well as the estimated overall total counts per administrative unit.
output_dir	This is the directory with the name of the output folder where the disaggregated population proportions and population totals are automatically saved.
class	This are the categories of the variables of interest. For example, for educational level, it could be 'no education', 'primary education', 'secondary education', 'tertiary education'.

### Value

A list of data frame objects of the output files including the disaggregated population proportions and population totals along with the corresponding measures of uncertainties (lower and upper bounds of 95-percent credible intervals) for each demographic characteristic. In addition, a file containing the model performance/model fit evaluation metrics is also produced.

### Examples

```
## Not run: data(toydata)
classes <- names(toydata$admin %>% dplyr::select(starts_with("age_")))
result2 <- spices(df = toydata$admin, output_dir = tempdir(), class = classes)
## End(Not run)
```

---

splash	<i>splash: Disaggregates population counts at high-resolution grid cells using building counts values of grid cells as a weighting layer. It is used for two-level disaggregation.</i>
--------	--

---

### Description

This function disaggregates population estimates at grid cell levels using the building counts of each grid cell to first disaggregate the admin unit's total population across the grid cells. Then, each grid cell's total count is further disaggregated into groups of interest using the admin's proportions.



## Usage

```
splash(df, rdf, rclass, output_dir)
```

## Arguments

df	A data frame object containing sample data (often partially observed) on different demographic groups population. It contains the admin's total population count to be disaggregated as well as other key variables as defined within the 'toydata'.
rdf	A gridded data frame object containing key information on the grid cells. Variables include the admin_id which must be identical to the one in the admin level data. It contains GPS coordinates. i.e, longitude (lon) and Latitude (lat) of the grid cell's centroids.
rclass	This is a user-defined names of the files to be saved in the output folder.
output_dir	This is the directory with the name of the output folder where the disaggregated population proportions and population totals are automatically saved.

## Value

A list of data frame objects of the output files including the disaggregated population proportions and population totals along with the corresponding measures of uncertainties (lower and upper bounds of 95-percent credible intervals) for each demographic characteristic. In addition, a file containing the model performance/model fit evaluation metrics is also produced.

## Examples

```
# run cheesepop
data(toydata)
result <- cheesepop(df = toydata$admin,output_dir = tempdir())
rclass <- paste0("TOY_population_v1_0_age",1:12) # Mean
result2 <- splash(df = result$full_data, rdf = toydata$grid, rclass, output_dir = tempdir())
ras2<- raster(paste0(output_dir = tempdir(), "/pop_TOY_population_v1_0_age4.tif"))
```

---

splash1	<i>splash1: Disaggregates population counts at high-resolution grid cells using building counts values of grid cells as a weighting layer. However, unlike 'splash' it is used for One-level disaggregation.</i>
---------	--

---

## Description

This function disaggregates population estimates at grid cell levels for one level of classification only. It uses the building counts of each grid cell to first disaggregate the admin unit's total population across the grid cells. Then, each grid cell's total count is further disaggregated into groups of interest using the admin's proportions.

## Usage

```
splash1(df, rdf, class, rclass, output_dir)
```

## Arguments

<code>df</code>	A data frame object containing sample data (often partially observed) on different demographic groups population. It contains the admin's total population count to be disaggregated as well as other key variables as defined within the 'toydata'.
<code>rdf</code>	A gridded data frame object containing key information on the grid cells. Variables include the <code>admin_id</code> which must be identical to the one in the admin level data. It contains GPS coordinates. i.e, longitude ( <code>lon</code> ) and Latitude ( <code>lat</code> ) of the grid cell's centroids.
<code>class</code>	These are the categories of the variables of interest. For example, for educational level, it could be 'no education', 'primary education', 'secondary education', 'tertiary education'.
<code>rclass</code>	This is a user-defined names of the files to be saved in the output folder.
<code>output_dir</code>	This is the directory with the name of the output folder where the disaggregated population proportions and population totals are automatically saved.

## Value

A list of data frame objects of the output files including the disaggregated population proportions and population totals along with the corresponding measures of uncertainties (lower and upper bounds of 95-percent credible intervals) for each demographic characteristic. In addition, a file containing the model performance/model fit evaluation metrics is also produced.

## Examples

```
data(toydata)
result <- cheesepop(df = toydata$admin, output_dir = tempdir()) # run cheesepop

rclass <- paste0("TOY_population_v1_0_age", 1:12) # Mean
class <- names(toydata$admin %>% dplyr::select(starts_with("age_")))
result2 <- splash1(df = result$full_data, rdf = toydata$grid, class, rclass, output_dir = tempdir())
ras2 <- raster(paste0(output_dir = tempdir(), "/pop_TOY_population_v1_0_age4.tif"))
```

---

<code>spray</code>	<i>Spray: Disaggregates population counts by dividing the admin total by the number of grid cells within the administrative units. Then admin proportions are used to further disaggregate the grid cell totals by groups.</i>
--------------------	--

---

## Description

This function disaggregates population estimates at grid cell levels when there are no information on the building and population counts.

## Usage

```
spray(df, rdf, rclass, output_dir)
```

## Arguments

df	A data frame object containing sample data (often partially observed) on different demographic groups population. It contains the admin's total population count to be disaggregated as well as other key variables as defined within the 'toydata'.
rdf	A gridded data frame object containing key information on the grid cells. Variables include the admin_id which must be identical to the one in the admin level data. It contains GPS coordinates. i.e, longitude (lon) and Latitude (lat) of the grid cell's centroids.
rclass	This is a user-defined names of the files to be saved in the output folder.
output_dir	This is the directory with the name of the output folder where the disaggregated population proportions and population totals are automatically saved.
class	These are the categories of the variables of interest. For example, for educational level, it could be 'no education', 'primary education', 'secondary education', 'tertiary education'.

## Value

A list of data frame objects of the output files including the disaggregated population proportions and population totals along with the corresponding measures of uncertainties (lower and upper bounds of 95-percent credible intervals) for each demographic characteristic. In addition, a file containing the model performance/model fit evaluation metrics is also produced.

## Examples

```
data(toydata)
result <- cheesepop(df = toydata$admin, output_dir = tempdir()) # run cheesepop

rclass <- paste0("TOY_population_v1_0_age", 1:12) # Mean
result2 <- spray(df = result$full_data, rdf = toydata$grid, rclass, output_dir = tempdir())
ras2 <- raster(paste0(output_dir = tempdir(), "/pop_TOY_population_v1_0_age4.tif"))
```

---

spray1	<i>spray1: Disaggregates population counts at high-resolution grid cells in the absence population and building counts - for one-level only</i>
--------	---

---

## Description

This function disaggregates population estimates at grid cell levels using the building counts of each grid cell to first disaggregate the admin unit's total population across the grid cells. Then, each grid cell's total count is further disaggregated into groups of interest using the admin's proportions.

## Usage

```
spray1(df, rdf, class, rclass, output_dir)
```

**Arguments**

df	A data frame object containing sample data (often partially observed) on different demographic groups population. It contains the admin's total population count to be disaggregated as well as other key variables as defined within the 'toydata'.
rdf	A gridded data frame object containing key information on the grid cells. Variables include the admin_id which must be identical to the one in the admin level data. It contains GPS coordinates. i.e, longitude (lon) and Latitude (lat) of the grid cell's centroids.
class	These are the categories of the variables of interest. For example, for educational level, it could be 'no education', 'primary education', 'secondary education', 'tertiary education'.
rclass	This is a user-defined names of the files to be saved in the output folder.
output_dir	This is the directory with the name of the output folder where the disaggregated population proportions and population totals are automatically saved.

**Value**

A list of data frame objects of the output files including the disaggregated population proportions and population totals along with the corresponding measures of uncertainties (lower and upper bounds of 95-percent credible intervals) for each demographic characteristic. In addition, a file containing the model performance/model fit evaluation metrics is also produced.

**Examples**

```
data(toydata)
result <- cheesepop(df = toydata$admin, output_dir = tempdir()) # run cheesepop

rclass <- paste0("TOY_population_v1_0_age", 1:12) # Mean
class <- names(toydata$admin %>% dplyr::select(starts_with("age_")))
result2 <- spray(df = result$full_data, rdf = toydata$grid, class, rclass, output_dir = tempdir())
ras2 <- raster(paste0(output_dir = tempdir(), "/pop_TOY_population_v1_0_age4.tif"))
```

---

sprinkle	<i>sprinkle: Disaggregates population counts at high-resolution grid cells using the grid cell's total population counts. Note that this could also be applied to more than two levels scenarios.</i>
----------	---

---

**Description**

This function disaggregates population estimates at grid cell levels using the population counts of each grid cell.

**Usage**

```
sprinkle(df, rdf, rclass, output_dir)
```

**Arguments**

df	A data frame object containing sample data (often partially observed) on different demographic groups population. It contains the admin's total population count to be disaggregated as well as other key variables as defined within the 'toydata'.
rdf	A gridded data frame object containing key information on the grid cells. Variables include the admin_id which must be identical to the one in the admin level data. It contains GPS coordinates. i.e, longitude (lon) and Latitude (lat) of the grid cell's centroids.
rclass	This is a user-defined names of the files to be saved in the output folder.
output_dir	This is the directory with the name of the output folder where the disaggregated population proportions and population totals are automatically saved.

**Value**

A list of data frame objects of the output files including the disaggregated population proportions and population totals along with the corresponding measures of uncertainties (lower and upper bounds of 95-percent credible intervals) for each demographic characteristic. In addition, a file containing the model performance/model fit evaluation metrics is also produced.

**Examples**

```
data(toydata)
result <- cheesepop(df = toydata$admin, output_dir = tempdir()) # run cheesepop

rclass <- paste0("TOY_population_v1_0_age", 1:12) # Mean
result2 <- spray(df = result$full_data, rdf = toydata$grid, rclass, output_dir = tempdir())
ras2 <- raster(paste0(output_dir = tempdir(), "/pop_TOY_population_v1_0_age4.tif"))
```

---

sprinkle1	<i>sprinkle1: Disaggregates population counts at high-resolution grid cells using the grid's total population. This can also be applied to one-level disaggregation.</i>
-----------	--

---

**Description**

This function disaggregates population estimates at grid cell levels using the population counts of each grid cell.

**Usage**

```
sprinkle1(df, rdf, class, rclass, output_dir)
```

**Arguments**

df	A data frame object containing sample data (often partially observed) on different demographic groups population. It contains the admin's total population count to be disaggregated as well as other key variables as defined within the 'toydata'.
----	--

rdf	A gridded data frame object containing key information on the grid cells. Variables include the admin_id which must be identical to the one in the admin level data. It contains GPS coordinates. i.e, longitude (lon) and Latitude (lat) of the grid cell's centroids.
class	These are the categories of the variables of interest. For example, for educational level, it could be 'no education', 'primary education', 'secondary education', 'tertiary education'.
rclass	This is a user-defined names of the files to be saved in the output folder.
output_dir	This is the directory with the name of the output folder where the disaggregated population proportions and population totals are automatically saved.

### Value

A list of data frame objects of the output files including the disaggregated population proportions and population totals along with the corresponding measures of uncertainties (lower and upper bounds of 95-percent credible intervals) for each demographic characteristic. In addition, a file containing the model performance/model fit evaluation metrics is also produced.

### Examples

```
data(toydata)
result <- cheesepop(df = toydata$admin, output_dir = tempdir()) # run cheesepop

rclass <- paste0("TOY_population_v1_0_age", 1:12) # Mean
result2 <- sprinkle1(df = result$full_data, rdf = toydata$grid, rclass, output_dir = tempdir())
ras2 <- raster(paste0(output_dir = tempdir(), "/pop_TOY_population_v1_0_age4.tif"))
```

---

toydata	<i>A list object containing two dataframes - an administrative-level dataset (admin) containing partially observed age-sex structured data, and a grid-cell level dataset (grid) for population disaggregation at 1km by 1km grid cells.</i>
---------	--

---

### Description

Artificially generated toy datasets that come in a cross-sectional format. The 'admin' data is a dataframe collated at administrative unit level which contains information on the observed number of individuals per age and sex groups within each administrative unit. Key variables include the administrative unit identifier (admin\_id), the admin total population to be disaggregated (total), the total number of buildings within each admin unit (bld), and the longitude (lon) and latitude (lat). The 'admin' data provides artificial information for 900 spatially distinct administrative units in which the individuals in the population are grouped into 12 mutually exclusive and exhaustive age groups. Each of the age groups was further grouped into 'male' and 'female' groups. The data contains the total population counts (total) for each spatial unit but also contains missing age and sex groups population counts. The model first predicts the population proportions of the missing data and then disaggregates the population totals using the predicted proportions to obtain the predicted population counts for the age and sex groups. Note that the same applies for other demographic groups such as marital status, race, etc.

### Usage

```
data(toydata)
```

## Format

An object of class "list"

**admin\_id** Available in both the 'admin' and 'grid' datasets. It is a numerical value which serves as the administrative units unique identifier. They should match perfectly for both the 'admin' and 'grid' datasets

**grid\_id** Available in the 'grid' dataset only. It is a numerical value which serves as the grid cell unique identifier.

**x1,x2,x3** These are the samples of geospatial covariates (only required for the 'cheesecake' and the 'slices' functions). Note that these are the covariates identified to significantly predict population distribution among the demographic groups. The package allows the user to include any number of covariates in their own datasets.

**total** Available in both the 'admin' and 'grid' datasets. It provides estimates of the total population counts to be disaggregated. It DOES NOT necessarily have to be a rowsum of the age groups totals.

**bld** Available in both the 'admin' and 'grid' datasets. It provides the total number of buildings in each grid cell or administrative unit.

**age\_1, ..., age\_12** These correspond to the partially or fully observed number of people for each age group. Note that only 12 age groups are used here for illustration purposes, however, the package can accommodate any number of age or sex or any demographic groups.

**fage\_1, ..., fage\_12** These correspond to the partially or fully observed number of females corresponding to each of the age groups. Note that only 12 age groups are used here for illustration purposes, however, the package can accommodate any number of age or sex or any demographic groups.

**mage\_1, ..., mage\_12** These correspond to the partially or fully observed number of males corresponding to each of the age groups. Note that only 12 age groups are used here for illustration purposes, however, the package can accommodate any number of age or sex or any demographic groups.

**lon** Available in both the 'admin' and 'grid' datasets. Provides the value of the longitude of the centroids of the grid cells or admin unit polygons.

**lat** Available in both the 'admin' and 'grid' datasets. Provides the value of the latitude of the centroids of the grid cells or admin unit polygons.

## Details

The second dataset in the toydata list is the 'grid' data which allows for the prediction of the age-sex structures at 1km by 1km grid cells (note that population predictions can be made at any spatial resolution of interest). The 'grid' data contains six key variables. These are administrative unit identifier (admin\_id) which must be identical to the those in the 'admin' data; the grid cell identifier (grid\_id); the total number of people per grid cell (total), if available; the total number of buildings per grid cell (bld), if available; and the longitude (lon) and latitude (lat) variables for the grid cell centroids.

illustrate the use of the package.

## References

This data set was artificially created for the purpose of illustrations within the jollofR package.

**Examples**

```
data(toydata)
head(toydata$admin)
head(toydata$grid)
```



# Index

## \* datasets

toydata, [14](#)

boxLine, [2](#)

cheesecake, [3](#)

cheesepop, [4](#)

plotHist, [5](#)

plotRast, [5](#)

pyramid, [6](#)

slices, [7](#)

spices, [8](#)

splash, [8](#)

splash1, [9](#)

spray, [10](#)

spray1, [11](#)

sprinkle, [12](#)

sprinkle1, [13](#)

toydata, [14](#)