

Package ‘SUMMER’

October 23, 2019

Type Package

Title Spatio-Temporal Under-Five Mortality Methods for Estimation

Version 0.3.0

Date 2019-10-18

Description Provides methods for estimating, projecting, and plotting spatio-temporal under-five mortality rates, described in Mercer et al. (2015) <doi:10.1214/15-AOAS872> and Li et al. (2019) <doi:10.1371/journal.pone.0210645>.

URL <https://github.com/bryandmartin/SUMMER>

BugReports <https://github.com/bryandmartin/SUMMER/issues>

Depends R (>= 3.5)

License GPL (>= 2)

Imports maptools, survey, stats, spdep, survival, ggplot2, utils,
Matrix, reshape2, viridis, sp

Encoding UTF-8

LazyData true

RoxygenNote 6.1.1

Additional_repositories <https://inla.r-inla-download.org/R/stable/>

Suggests INLA, knitr, rmarkdown, readstata13, gridExtra, R.rsp

VignetteBuilder R.rsp, knitr

NeedsCompilation no

Author Bryan D Martin [aut],
Zehang R Li [cre, aut],
Yuan Hsiao [aut],
Jessica Godwin [aut],
Jon Wakefield [aut],
Samuel J Clark [aut],
Geir-Arne Fuglstad [aut],
Andrea Riebler [aut]

Maintainer Zehang R Li <lizehang@gmail.com>

Repository CRAN

Date/Publication 2019-10-23 07:40:07 UTC

R topics documented:

| | |
|---------------------------|----|
| SUMMER-package | 3 |
| aggregateSurvey | 3 |
| BRFSS | 4 |
| ChangeRegion | 4 |
| DemoData | 5 |
| DemoData2 | 6 |
| DemoMap | 6 |
| DemoMap2 | 7 |
| expit | 7 |
| fitGeneric | 8 |
| fitINLA | 10 |
| fitINLA2 | 12 |
| getAdjusted | 14 |
| getAmat | 15 |
| getBirths | 16 |
| getCounts | 18 |
| getDiag | 19 |
| getDirect | 20 |
| getDirectList | 21 |
| getSmoothed | 22 |
| hatchPlot | 24 |
| iid.new | 26 |
| iid.new.pc | 26 |
| KenData | 27 |
| KingCounty | 27 |
| logit | 28 |
| mapPlot | 28 |
| mapPoints | 30 |
| plot.SUMMERproj | 31 |
| read_shape | 33 |
| rst | 33 |
| rw.new | 35 |
| rw.new.pc | 35 |
| simhyper | 36 |
| st.new | 37 |
| st.new.pc | 37 |

SUMMER-package

SUMMER package documentation.

Description

SUMMER provides methods for estimating, projecting, and plotting spatio-temporal under-five mortality rates.

Details

For details on the model implemented in this package, see Mercer et al. (2015) <doi:10.1214/15-AOAS872>.

The development version of the package will be maintained on <https://github.com/bryandmartin/SUMMER>.

aggregateSurvey

Aggregate estimators from different surveys.

Description

Aggregate estimators from different surveys.

Usage

```
aggregateSurvey(data)
```

Arguments

data Output from [getDirectList](#)

Value

Estimators aggregated across surveys.

Examples

```
## Not run:
data(DemoData)
data(DemoMap)
years <- levels(DemoData[[1]]$time)

# obtain direct estimates
data <- getDirectList(births = DemoData,
years = years,
regionVar = "region", timeVar = "time",
clusterVar = "~clustid+id",
```

```

ageVar = "age", weightsVar = "weights",
geo.recode = NULL)

# obtain maps
geo <- DemoMap$geo
mat <- DemoMap$Amat

# Simulate hyper priors
priors <- simhyper(R = 2, nsamp = 1e+05, nsamp.check = 5000, Amat = mat, only.iid = TRUE)

# combine data from multiple surveys
data <- aggregateSurvey(data)
utils::head(data)

## End(Not run)

```

BRFSS

The BRFSS dataset

Description

The Behavioral Risk Factor Surveillance System (BRFSS) is an annual telephone health survey conducted by the Centers for Disease Control and Prevention (CDC) that tracks health conditions and risk behaviors in the United States and its territories since 1984. This BRFSS dataset contains 16124 observations. The ‘diab2’ variable is the binary indicator of Type II diabetes, ‘strata’ is the strata indicator and ‘rwt_llcp’ is the final design weight. Records with missing HRA code or diabetes status are removed from this dataset. See http://www.cdc.gov/brfss/annual_data/2013/pdf/Weighting_Data.pdf for more details of the weighting procedure.

Usage

```
data(BRFSS)
```

Format

A data.frame of 26 variables.

ChangeRegion

Function to map region names to a common set.

Description

Function to map region names to a common set.

Usage

```
ChangeRegion(data, Bmat, regionVar = "region")
```

Arguments

| | |
|-----------|---|
| data | Preprocessed data |
| Bmat | Matrix of changes. Each row corresponds to a region name possibly in the data files, and each column corresponds to a region after mapping. The values in the matrix are binary. The row names and column names need to be specified to the region names. |
| regionVar | String indicating the region variable. Defaults to 'region'. |

Value

Data after changing region names

Examples

```
# Construct a small test data
testdata <- data.frame(region = c("north", "south", "east",
  "south", "east"), index = c(1:5))

# Construct a changing rule: combining south and east
Bmat <- matrix(c(1, 0, 0, 0, 1, 1), 3, 2)
colnames(Bmat) <- c("north", "south and east")
rownames(Bmat) <- c("north", "south", "east")
print(Bmat)

# New data after transformation
test <- ChangeRegion(testdata, Bmat, "region")
print(test)
```

DemoData

Fake dataset for vignette.

Description

A small fake dataset with 4 regions and 5 survey years. This does not represent any real country's data and are based on a subset of the model dataset provided by DHS.

Usage

```
data(DemoData)
```

Format

A list of with five components, named by survey year.

Source

<https://dhsprogram.com/data/model-datasets.cfm>

DemoData2

Fake dataset for vignette.

Description

A small fake dataset with 8 regions and two response variables: age and tobacco.use. This does not represent any real country's data and are based on a subset of the model dataset provided by DHS.

Usage

```
data(DemoData2)
```

Format

A data.frame of 7 variables.

Source

<https://dhsprogram.com/data/model-datasets.cfm>

DemoMap

Map dataset for vignette.

Description

Shapefiles are from 1995 Uganda Admin 1 regions provided by DHS, but the data do not represent real information about any country.

Usage

```
data(DemoMap)
```

Format

An object of class list of length 2.

Details

- geo. Geographic map files
- Amat. Adjacency matrix for regions

Source

<https://spatialdata.dhsprogram.com/boundaries/#view=table&countryId=UG>

DemoMap2

Map dataset for vignette.

Description

Shapefiles are from 2014 Kenya Admin 1 regions provided by DHS, but the data do not represent real information about any country.

Usage

```
data(DemoMap2)
```

Format

An object of class `list` of length 2.

Details

- `geo`. Geographic map files
- `Amat`. Adjacency matrix for regions

Source

<https://spatialdata.dhsprogram.com/boundaries/#view=table&countryId=KE>

expit

Expit transformation

Description

Expit transformation

Usage

```
expit(x)
```

Arguments

x data

Value

expit of x

Examples

```
x <- .5
expit(x)
```

| | |
|------------|--|
| fitGeneric | <i>Fit space-time smoothing models for a generic outcome from complex surveys.</i> |
|------------|--|

Description

This function calculates the direct estimates by region and fit a simple spatial smoothing model to the direct estimates adjusting for survey design.

Usage

```
fitGeneric(data, geo, Amat, X = NULL, responseType = c("binary",
  "gaussian")[1], responseVar, strataVar = "strata",
  weightVar = "weights", regionVar = "region",
  clusterVar = "~v001+v002", pc.u = 1, pc.alpha = 0.01,
  pc.u.phi = 0.5, pc.alpha.phi = 2/3, CI = 0.95, FUN = NULL,
  formula = NULL, timeVar = NULL, time.model = c("rw1", "rw2")[1],
  type.st = 1)
```

Arguments

| | |
|--------------|--|
| data | data frame with region and strata information. |
| geo | Geo file |
| Amat | Adjacency matrix for the regions |
| X | Covariate matrix with the first column being the region names. Currently only supporting static region-level covariates. |
| responseType | Type of the response variable, currently supports 'binary' (default with logit link function) or 'gaussian'. |
| responseVar | the response variable |
| strataVar | the strata variable |
| weightVar | the weights variable |
| regionVar | Variable name for region, typically 'v024', for older surveys might be 'v101' |
| clusterVar | Variable name for cluster, typically '~v001 + v002' |
| pc.u | hyperparameter U for the PC prior on precisions. |
| pc.alpha | hyperparameter alpha for the PC prior on precisions. |
| pc.u.phi | hyperparameter U for the PC prior on the mixture probability phi in BYM2 model. |

| | |
|--------------|--|
| pc.alpha.phi | hyperparameter alpha for the PC prior on the mixture probability phi in BYM2 model. |
| CI | the desired posterior credible interval to calculate |
| FUN | the function to transform the posterior draws. Default to be identify function for normal variable and inverse logit transformation for binomial variables |
| formula | a string of user-specified random effects model to be used in the INLA call |
| timeVar | The variable indicating time period. If set to NULL then the temporal model and space-time interaction model are ignored. |
| time.model | the model for temporal trends and interactions. It can be either "rw1" or "rw2". |
| type.st | can take values 0 (no interaction), or 1 to 4, corresponding to the type I to IV space-time interaction. |

Details

Normal or binary variables are currently supported. For binary variables, the logit transformation is performed on the direct estimates of probabilities, and a Gaussian additive model is fitted on the logit scale using INLA.

Value

| | |
|--------------|------------------------------|
| HT | Direct estimates |
| smooth | Spatially smoothed estimates |
| fit | a fitted INLA object |
| geo | input argument |
| Amat | input argument |
| CI | input argument |
| responseType | input argument |
| FUN | input argument |

See Also

[getDirectList](#), [fitINLA](#)

Examples

```
## Not run:
data(DemoData2)
data(DemoMap2)
fit <- fitGeneric(data=DemoData2, geo=DemoMap2$geo,
  Amat=DemoMap2$Amat, responseType="binary",
  responseVar="tobacco.use", strataVar="strata",
  weightVar="weights", regionVar="region",
  clusterVar = "~clustid+id", CI = 0.95)

# Example with region-level covariates
Xmat <- aggregate(age~region, data = DemoData2, FUN = mean)
```

```
fit <- fitGeneric(data=DemoData2, geo=DemoMap2$geo,
  Amat=DemoMap2$Amat, responseType="binary",
  X = Xmat,
  responseVar="tobacco.use", strataVar="strata",
  weightVar="weights", regionVar="region",
  clusterVar = "~clustid+id", CI = 0.95)
```

```
## End(Not run)
```

 fitINLA

Fit space-time smoothing models to mortality rates

Description

Fit space-time smoothing models to mortality rates

Usage

```
fitINLA(data, Amat, geo, formula = NULL, rw = 2, is.yearly = TRUE,
  year_label, year_range = c(1980, 2014), m = 5, na.rm = TRUE,
  priors = NULL, type.st = 1, hyper = c("pc", "gamma")[1],
  pc.u = 1, pc.alpha = 0.01, pc.u.phi = 0.5, pc.alpha.phi = 2/3,
  a.iid = NULL, b.iid = NULL, a.rw = NULL, b.rw = NULL,
  a.icar = NULL, b.icar = NULL, options = list(dic = T, mlik = T, cpo
  = T, openmp.strategy = "default"), verbose = FALSE)
```

Arguments

| | |
|------------|--|
| data | Combined dataset |
| Amat | Adjacency matrix for the regions |
| geo | Geo file |
| formula | INLA formula. See vignette for example of using customized formula. |
| rw | Take values 1 or 2, indicating the order of random walk. |
| is.yearly | Logical indicator for fitting yearly or period model. |
| year_label | string vector of year names |
| year_range | Entire range of the years (inclusive) defined in year_label. |
| m | Number of years in each period. |
| na.rm | Logical indicator of whether to remove rows with NA values in the data. Default set to TRUE. |
| priors | priors from simhyper |
| type.st | type for space-time interaction |
| hyper | which hyperpriors to use. Default to be using the PC prior ("pc"). |
| pc.u | hyperparameter U for the PC prior on precisions. |

| | |
|--------------|---|
| pc.alpha | hyperparameter alpha for the PC prior on precisions. |
| pc.u.phi | hyperparameter U for the PC prior on the mixture probability phi in BYM2 model. |
| pc.alpha.phi | hyperparameter alpha for the PC prior on the mixture probability phi in BYM2 model. |
| a.iid | hyperparameter for i.i.d random effects. |
| b.iid | hyperparameter for i.i.d random effects. |
| a.rw | hyperparameter for RW 1 or 2 random effects. |
| b.rw | hyperparameter for RW 1 or 2 random effects. |
| a.icar | hyperparameter for ICAR random effects. |
| b.icar | hyperparameter for ICAR random effects. |
| options | list of options to be passed to control.compute() in the inla() function. |
| verbose | logical indicator to print out detailed inla() intermediate steps. |

Value

INLA model fit using the provided formula, country summary data, and geographic data

See Also

[getDirect](#)

Examples

```
## Not run:
years <- levels(DemoData[[1]]$time)

# obtain direct estimates
data <- getDirectList(births = DemoData,
  years = years,
  regionVar = "region", timeVar = "time",
  clusterVar = "~clustid+id",
  ageVar = "age", weightsVar = "weights",
  geo.recode = NULL)
# obtain direct estimates
data_multi <- getDirectList(births = DemoData, years = years,
  regionVar = "region", timeVar = "time", clusterVar = "~clustid+id",
  ageVar = "age", weightsVar = "weights", geo.recode = NULL)
data <- aggregateSurvey(data_multi)

# national model
years.all <- c(years, "15-19")
fit1 <- fitINLA(data = data, geo = NULL, Amat = NULL,
  year_label = years.all, year_range = c(1985, 2019),
  rw = 2, is.yearly=FALSE, m = 5)
out1 <- getSmoothed(fit1)
plot(out1, is.subnational=FALSE)
```

```
# subnational model
fit2 <- fitINLA(data = data, geo = geo, Amat = mat,
  year_label = years.all, year_range = c(1985, 2019),
  rw = 2, is.yearly=TRUE, m = 5, type.st = 4)
out2 <- getSmoothed(fit2, Amat = mat)
plot(out2, is.yearly=TRUE, is.subnational=TRUE)
```

```
## End(Not run)
```

fitINLA2

Fit cluster-level space-time smoothing models to mortality rates

Description

Fit cluster-level space-time smoothing models to mortality rates

Usage

```
fitINLA2(data, family = c("betabinomial", "binomial")[1],
  age.groups = c("0", "1-11", "12-23", "24-35", "36-47", "48-59"),
  age.n = c(1, 11, 12, 12, 12, 12), age.rw.group = 1:6, Amat, geo,
  bias.adj = NULL, bias.adj.by = NULL, formula = NULL, rw = 2,
  year_label, priors = NULL, type.st = 1, hyper = c("pc",
  "gamma")[1], pc.u = 1, pc.alpha = 0.01, pc.u.phi = 0.5,
  pc.alpha.phi = 2/3, a.iid = NULL, b.iid = NULL, a.rw = NULL,
  b.rw = NULL, a.icar = NULL, b.icar = NULL, options = list(config
  = TRUE), verbose = FALSE)
```

Arguments

| | |
|------------|--|
| data | count data of person-months with the following columns <ul style="list-style-type: none"> • cluster: cluster ID • years: time period • region: region of the cluster • strata: stratum of the cluster • age: age group corresponding to the row • total: total number of person-month in this age group, stratum, cluster, and period • Y: total number of deaths in this age group, stratum, cluster, and period |
| family | family of the model. This can be either binomial (with logistic normal prior) or betabinomial. |
| age.groups | a character vector of age groups in increasing order. |
| age.n | number of months in each age groups in the same order. |

| | |
|--------------|---|
| age.rw.group | vector indicating grouping of the ages groups. For example, if each age group is assigned a different random walk component, then set age.rw.group to <code>c(1:length(age.groups))</code> ; if all age groups share the same random walk component, then set age.rw.group to a <code>rep(1, length(age.groups))</code> . The default for 6 age groups is <code>c(1,2,3,3,3,3)</code> , which assigns a separate random walk to the first two groups and a common random walk for the rest of the age groups. The vector should contain values starting from 1. |
| Amat | Adjacency matrix for the regions |
| geo | Geo file |
| bias.adj | the ratio of unadjusted mortality rates or age-group-specific hazards to the true rates or hazards. It needs to be a data frame that can be merged to the outcome, i.e., with the same column names for time periods (for national adjustment), or time periods and region (for subnational adjustment). The column specifying the adjustment ratio should be named "ratio". |
| bias.adj.by | vector of the column names specifying how to merge the bias adjustment to the count data. For example, if bias adjustment factor is provided in bias.adj for each region and time, then bias.adj.by should be <code>'c("region", "time")'</code> . |
| formula | INLA formula. See vignette for example of using customized formula. |
| rw | Take values 1 or 2, indicating the order of random walk. |
| year_label | string vector of year names |
| priors | priors from simhyper |
| type.st | type for space-time interaction |
| hyper | which hyperpriors to use. Default to be using the PC prior ("pc"). |
| pc.u | hyperparameter U for the PC prior on precisions. |
| pc.alpha | hyperparameter alpha for the PC prior on precisions. |
| pc.u.phi | hyperparameter U for the PC prior on the mixture probability phi in BYM2 model. |
| pc.alpha.phi | hyperparameter alpha for the PC prior on the mixture probability phi in BYM2 model. |
| a.iid | hyperparameter for i.i.d random effects. |
| b.iid | hyperparameter for i.i.d random effects. |
| a.rw | hyperparameter for RW 1 or 2 random effects. |
| b.rw | hyperparameter for RW 1 or 2 random effects. |
| a.icar | hyperparameter for ICAR random effects. |
| b.icar | hyperparameter for ICAR random effects. |
| options | list of options to be passed to <code>control.compute()</code> in the <code>inla()</code> function. |
| verbose | logical indicator to print out detailed <code>inla()</code> intermediate steps. |

Value

INLA model fit using the provided formula, country summary data, and geographic data

See Also[getDirect](#)**Examples**

```
message("Please check the package vignette on binomial models.")
```

| | |
|--------------------------|---|
| <code>getAdjusted</code> | <i>Adjust direct estimates and their associated variances</i> |
|--------------------------|---|

Description

Adjust direct estimates and their associated variances

Usage

```
getAdjusted(data, ratio, time = "years", region = "region",
  est = "mean", logit = "logit.est", logit.var = "var.est",
  logit.prec = "logit.prec", logit.lower = "lower",
  logit.upper = "upper", prob.lower = NULL, prob.upper = NULL,
  adj = "ratio", verbose = FALSE, lower = NULL, upper = NULL)
```

Arguments

| | |
|--------------------------|--|
| <code>data</code> | data frame of the adjusted estimates and the associated uncertainties, see the arguments below for specific columns. |
| <code>ratio</code> | the ratio of unadjusted mortality rates to the true mortality rates. It can be either a data frame with the following three columns (region, time, and adj) if adjustment factor differ by region; or a data frame with the following two columns (time and adj) if adjustment factor only varies over time. The column names specifying region, time and ratio, and adjustment are specified by the arguments in the function call. |
| <code>time</code> | the column name for time in the data and adjustment ratio. |
| <code>region</code> | the column name for region in the data and adjustment ratio. |
| <code>est</code> | the column name for unadjusted mortality rates in the data |
| <code>logit</code> | the column name for the logit of the unadjusted mortality rates in the data |
| <code>logit.var</code> | the column name for the variance of the logit of the unadjusted mortality rates in the data |
| <code>logit.prec</code> | the column name for the precision of the logit of the unadjusted mortality rates in the data |
| <code>logit.lower</code> | the column name for the 95% lower bound of the logit of the unadjusted mortality rates in the data |
| <code>logit.upper</code> | the column name for the 95% lower bound of the logit of the unadjusted mortality rates in the data |

| | |
|------------|---|
| prob.lower | the column name for the 95% lower bound of the unadjusted mortality rates in the data. If this is provided instead of logit.lower, the logit scale lower bound will be created. |
| prob.upper | the column name for the 95% upper bound of the unadjusted mortality rates in the data. If this is provided instead of logit.upper, the logit scale upper bound will be created. |
| adj | the column name for the adjustment ratio |
| verbose | logical indicator for whether to print out unadjusted row index |
| lower | previous argument name for prob.lower. Will be removed in the next update |
| upper | previous argument name for prob.upper. Will be removed in the next update |

Value

adjusted dataset of the same columns.

Examples

```
## Not run:
years <- levels(DemoData[[1]]$time)

# obtain direct estimates
data <- getDirectList(births = DemoData,
  years = years,
  regionVar = "region", timeVar = "time",
  clusterVar = "~clustid+id",
  ageVar = "age", weightsVar = "weights",
  geo.recode = NULL)
# obtain direct estimates
data_multi <- getDirectList(births = DemoData, years = years,
  regionVar = "region", timeVar = "time", clusterVar = "~clustid+id",
  ageVar = "age", weightsVar = "weights", geo.recode = NULL)
data <- aggregateSurvey(data_multi)

# randomly simulate adjustment factor
adj <- expand.grid(region = unique(data$region), years = years)
adj$ratio <- runif(dim(adj)[1], min = 0.5, max = 0.8)
data.adj <- getAdjusted(data = data, ratio = adj)

## End(Not run)
```

getAmat

Extract adjacency matrix from the map

Description

Extract adjacency matrix from the map

Usage

```
getAmat(geo, names)
```

Arguments

| | |
|-------|---|
| geo | SpatialPolygonsDataFrame of the map |
| names | character vector of region ids to be added to the neighbours list |

Value

Spatial djacency matrix.

Examples

```
data(DemoMap)
mat <- getAmat(geo = DemoMap$geo, names = DemoMap$geo$REGNAME)
mat
DemoMap$Amat
```

getBirths

Function to get Births file from DHS .dta files.

Description

Function to get Births file from DHS .dta files.

Usage

```
getBirths(filepath = NULL, data = NULL, surveyyear = NA,
  variables = c("caseid", "v001", "v002", "v004", "v005", "v021", "v022",
    "v023", "v024", "v025", "v139", "bidx"), strata = c("v024", "v025"),
  dob = "b3", alive = "b5", age = "b7", date.interview = "v008",
  month.cut = c(1, 12, 24, 36, 48, 60), year.cut = seq(1980, 2020, by =
    5), cmc.adjust = 0, compact = FALSE, compact.by = c("v001", "v024",
    "v025", "v005"))
```

Arguments

| | |
|------------|---|
| filepath | file path of raw .dta file from DHS. Only used when data frame is not provided in the function call. |
| data | data frame of a DHS survey |
| surveyyear | year of survey |
| variables | vector of variables to be used in obtaining the person-month files. The variables correspond the the DHS recode manual VI. For early DHS data, the variable names may need to be changed. |

| | |
|-----------------------------|---|
| <code>strata</code> | vector of variable names used for strata. If a single variable is specified, then that variable will be used as strata indicator. If multiple variables are specified, the interaction of these variables will be used as strata indicator. |
| <code>dob</code> | variable name for the date of birth. |
| <code>alive</code> | variable name for the indicator of whether child was alive or dead at the time of interview. |
| <code>age</code> | variable name for the age at death of the child in completed months. |
| <code>date.interview</code> | variable name for the date of interview. |
| <code>month.cut</code> | the cutoff of each bins of age group in the unit of months. Default values are 1, 12, 24, 36, 48, and 60, representing the age groups (0, 1), [1, 12), [12, 24), ..., [48, 60). |
| <code>year.cut</code> | The cutoff of each bins of time periods, including both boundaries. Default values are 1980, 1985, ..., 2020, representing the time periods 80-84, 85-89, ..., 15-19. Notice that if each bin contains one year, the last year in the output is <code>max(year.cut)-1</code> . For example, if <code>year.cut = 1980:2020</code> , the last year in the output is 2019. |
| <code>cmc.adjust</code> | number of months to add to the recorded month in the dataset. Some DHS surveys does not use Gregorian calendar (the calendar used in most of the world). For example, the Ethiopian calendar is 92 months behind the Gregorian calendar in general. Then we can set <code>cmc.adjust</code> to 92, which adds 92 months to all dates in the dataset, effectively transforming the Ethiopian calendar to the Gregorian calendar. |
| <code>compact</code> | logical indicator of whether the compact format is returned. In the compact output, person months are aggregated by cluster, age, and time. Total number of person months and deaths in each group are returned instead of the raw person-months. |
| <code>compact.by</code> | vector of variables to summarize the compact form by. |

Value

This function returns a new data frame where each row indicate a person-month, with the additional variables specified in the function argument.

Examples

```
## Not run:
my_fp <- "/myExampleFilepath/surveyData.DTA"
DemoData <- getBirths(filepath = my_fp, surveyyear = 2015)

## End(Not run)
```

 getCounts

Aggregate person-month data into counts and totals by groups.

Description

Aggregate person-month data into counts and totals by groups.

Usage

```
getCounts(data, variables, by, ignore = NULL, addtotal = TRUE)
```

Arguments

| | |
|-----------|---|
| data | dataset in person-month format |
| variables | a character vector of the variables to aggregate |
| by | a character vector of columns that specifies which groups to aggregate by. |
| ignore | list of conditions not to impute 0. If left unspecified, any group levels not in the data will be imputed to have 0 counts. |
| addtotal | logical indicator of whether to add a column of group total counts. |

Value

data.frame of the ggregated counts.

Examples

```
# a toy dataset with 4 time periods but one missing in data
timelist <- factor(1:4)
data = data.frame(died = c(0,0,0,1,1,0,0),
  area = c(rep(c("A", "B"), 3), "A"),
  time = timelist[c(1,1,2,3,3,3,3)])
data
# without ignore argument, all levels will be imputed
getCounts(data, variables = "died", by = c("area", "time"))

# ignoring time = 4, the ignored level will not be imputed (but still in the output)
getCounts(data, variables = "died", by = c("area", "time"),
  ignore = list("time"=c(4)) )
```

| | |
|---------|------------------------------|
| getDiag | <i>Make diagnostic plots</i> |
|---------|------------------------------|

Description

Make diagnostic plots

Usage

```
getDiag(inla_mod, field = c("space", "time", "spacetime")[1],
        year_range = c(1985, 2019), year_label = c("85-89", "90-94", "95-99",
        "00-04", "05-09", "10-14", "15-19"), Amat = NULL, CI = 0.95)
```

Arguments

| | |
|------------|---|
| inla_mod | output from fitINLA |
| field | which random effects to plot. It can be one of the following: space, time, and spacetime. |
| year_range | Entire range of the years (inclusive) defined in year_label. To be deprecated and imputed from the fitted object in the next version of SUMMER. |
| year_label | vector of year string vector |
| Amat | adjacency matrix |
| CI | Desired level of credible intervals |

Value

List of diagnostic plots

Examples

```
## Not run:
data(DemoMap)
years <- levels(DemoData[[1]]$time)

# obtain direct estimates
data <- getDirectList(births = DemoData,
years = years,
regionVar = "region", timeVar = "time",
clusterVar = "~clustid+id",
ageVar = "age", weightsVar = "weights",
geo.recode = NULL)
# obtain direct estimates
data_multi <- getDirectList(births = DemoData, years = years,
regionVar = "region", timeVar = "time", clusterVar = "~clustid+id",
ageVar = "age", weightsVar = "weights", geo.recode = NULL)
data <- aggregateSurvey(data_multi)
```

```

# national model
years.all <- c(years, "15-19")
fit1 <- fitINLA(data = data, geo = DemoMap$geo, Amat = DemoMap$Amat,
  year_label = years.all, year_range = c(1985, 2019),
  rw = 2, is.yearly=FALSE, m = 5)
random.time <- getDiag(fit1, field = "time", year_label = years.all, #' year_range = c(1985, 2019))
random.space <- getDiag(fit1, field = "space", Amat = DemoMap$Amat)
random.spacetime <- getDiag(fit1, field = "spacetime",
  year_label = years, year_range = c(1985, 2019),
  Amat = DemoMap$Amat)

## End(Not run)

```

| | |
|-----------|---|
| getDirect | <i>Obtain the Horvitz-Thompson direct estimates and standard errors using delta method for a single survey.</i> |
|-----------|---|

Description

Obtain the Horvitz-Thompson direct estimates and standard errors using delta method for a single survey.

Usage

```

getDirect(births, years, regionVar = "region", timeVar = "time",
  clusterVar = "~v001+v002", ageVar = "age", weightsVar = "v005",
  Ntrials = NULL, geo.recode = NULL, national.only = FALSE)

```

Arguments

| | |
|---------------|---|
| births | A matrix child-month data from getBirths |
| years | String vector of the year intervals used |
| regionVar | Variable name for region in the input births data. |
| timeVar | Variable name for the time period indicator in the input births data. |
| clusterVar | Variable name for cluster, typically '~v001 + v002' |
| ageVar | Variable name for age group. This variable need to be in the form of "a-b" where a and b are both ages in months. For example, "1-11" means age between 1 and 11 months, including both end points. An exception is age less than one month can be represented by "0" or "0-0". |
| weightsVar | Variable name for sampling weights, typically 'v005' |
| Ntrials | Variable for the total number of person-months if the input data (births) is in the compact form. |
| geo.recode | The recode matrix to be used if region name is not consistent across different surveys. See ChangeRegion . |
| national.only | Logical indicator to obtain only the national estimates |

Value

a matrix of period-region summary of the Horvitz-Thompson direct estimates by region and time period specified in the argument, the standard errors using delta method for a single survey, the 95% confidence interval, and the logit of the estimates.

See Also

[getDirectList](#)

Examples

```
## Not run:
data(DemoData)
years <- c("85-89", "90-94", "95-99", "00-04", "05-09", "10-14")
mean <- getDirect(births = DemoData[[1]], years = years,
  regionVar = "region", timeVar = "time", clusterVar = "~clustid+id",
  ageVar = "age", weightsVar = "weights", geo.recode = NULL)

## End(Not run)
```

| | |
|---------------|--|
| getDirectList | <i>Obtain the Horvitz-Thompson direct estimates and standard errors using delta method for multiple surveys.</i> |
|---------------|--|

Description

Obtain the Horvitz-Thompson direct estimates and standard errors using delta method for multiple surveys.

Usage

```
getDirectList(births, years, regionVar = "region", timeVar = "time",
  clusterVar = "~v001+v002", ageVar = "age", weightsVar = "v005",
  Ntrials = NULL, geo.recode = NULL, national.only = FALSE)
```

Arguments

| | |
|------------|---|
| births | A list of child-month data from multiple surveys from getBirths . The name of the list is used as the identifier in the output. |
| years | String vector of the year intervals used |
| regionVar | Variable name for region, typically 'v024', for older surveys might be 'v101' |
| timeVar | Variable name for the time period indicator in the input births data. |
| clusterVar | Variable name for the IDs in the second-stage cluster sampling, typically '~v001 + v002', i.e., the cluster number and household number. When no cluster sampling design exists, this variable usually is the household ID. |

| | |
|---------------|---|
| ageVar | Variable name for age group. This variable need to be in the form of "a-b" where a and b are both ages in months. For example, "1-11" means age between 1 and 11 months, including both end points. An exception is age less than one month can be represented by "0" or "0-0". |
| weightsVar | Variable name for sampling weights, typically 'v005' |
| Ntrials | Variable for the total number of person-months if the input data (births) is in the compact form. |
| geo.recode | The recode matrix to be used if region name is not consistent across different surveys. See ChangeRegion . |
| national.only | Logical indicator to obtain only the national estimates |

Value

This is the extension to the [getDirect](#) function that returns estimates from multiple surveys. Additional columns in the output (survey and surveyYears) specify the estimates from different surveys.

See Also

[getDirect](#)

Examples

```
## Not run:
data(DemoData)
years <- c("85-89", "90-94", "95-99", "00-04", "05-09", "10-14")
mean <- getDirectList(births = DemoData, years = years,
  regionVar = "region", timeVar = "time", clusterVar = "~clustid+id",
  ageVar = "age", weightsVar = "weights", geo.recode = NULL)

## End(Not run)
```

| | |
|-------------|---|
| getSmoothed | <i>Function to obtain projected estimates from INLA for each time and region.</i> |
|-------------|---|

Description

Function to obtain projected estimates from INLA for each time and region.

Usage

```
getSmoothed(inla_mod, year_range = c(1985, 2019),
  year_label = c("85-89", "90-94", "95-99", "00-04", "05-09", "10-14",
    "15-19"), Amat = NULL, nsim = 1000, weight.strata = NULL,
  verbose = FALSE, mc = 0, include_time_unstruct = FALSE, ...)
```

Arguments

| | |
|-----------------------|--|
| inla_mod | output from fitINLA |
| year_range | range corresponding to year label |
| year_label | vector of year string vector |
| Amat | adjacency matrix |
| nsim | number of simulations |
| weight.strata | a data frame with three columns, years, region, and proportion of each strata for the corresponding time period and region. |
| verbose | logical indicator whether to print progress messages from inla.posterior.sample. |
| mc | number of monte carlo draws to approximate the marginal prevalence/hazards for binomial model. If mc = 0, analytical approximation is used. The analytical approximation is invalid for hazard modeling with more than one age groups. |
| include_time_unstruct | logical indicator whether to include the temporal unstructured effects (i.e., shocks) in the smoothed estimates. |
| ... | additional configurations passed to inla.posterior.sample. |

Value

Results from RW2 model fit, including projection.

See Also

[plot.SUMMERproj](#)

Examples

```
## Not run:
years <- levels(DemoData[[1]]$time)

# obtain direct estimates
data <- getDirectList(births = DemoData,
  years = years,
  regionVar = "region", timeVar = "time",
  clusterVar = "~clustid+id",
  ageVar = "age", weightsVar = "weights",
  geo.recode = NULL)
# obtain direct estimates
data_multi <- getDirectList(births = DemoData, years = years,
  regionVar = "region", timeVar = "time", clusterVar = "~clustid+id",
  ageVar = "age", weightsVar = "weights", geo.recode = NULL)
data <- aggregateSurvey(data_multi)

# national model
years.all <- c(years, "15-19")
fit1 <- fitINLA(data = data, geo = NULL, Amat = NULL,
  year_label = years.all, year_range = c(1985, 2019),
  rw = 2, is.yearly=FALSE, m = 5)
```

```

out1 <- getSmoothed(fit1)
plot(out1, is.subnational=FALSE)

# subnational model
fit2 <- fitINLA(data = data, geo = geo, Amat = mat,
  year_label = years.all, year_range = c(1985, 2019),
  rw = 2, is.yearly=TRUE, m = 5, type.st = 4)
out2 <- getSmoothed(fit2, Amat = mat)
plot(out2, is.yearly=TRUE, is.subnational=TRUE)

## End(Not run)

```

hatchPlot

Makes map plot with uncertainty hatching.

Description

This function visualizes the map with different variables. The input data frame can be either the long or wide format.

Usage

```

hatchPlot(data, variables, values = NULL, labels = NULL, geo, by.data,
  by.geo, is.long = FALSE, lower, upper, lim = NULL, lim.CI = NULL,
  breaks.CI = NULL, ncol = 4, hatch = NULL, border = NULL,
  size = 1, legend.label = NULL, per1000 = FALSE, ...)

```

Arguments

| | |
|-----------|--|
| data | a data frame with variables to be plotted |
| variables | vector of variables to be plotted. If long format of data is used, only one variable can be selected |
| values | the column corresponding to the values to be plotted, only used when long format of data is used |
| labels | vector of labels to use for each variable, only used when wide format of data is used |
| geo | geo output from read_shape |
| by.data | column name specifying region names in the data |
| by.geo | variable name specifying region names in the data |
| is.long | logical indicator of whether the data is in the long format, default to FALSE |
| lower | column name of the lower bound of the CI |
| upper | column name of the upper bound of the CI |

| | |
|--------------|--|
| lim | fixed range of values for the variables to plot |
| lim.CI | fixed range of the CI widths to plot |
| breaks.CI | a vector of numerical values that decides the breaks in the CI widths to be shown |
| ncol | number of columns for the output tabs |
| hatch | color of the hatching lines. |
| border | color of the polygon borders. |
| size | line width of the polygon borders. |
| legend.label | Label for the color legend. |
| per1000 | logical indicator to plot mortality rates as rates per 1,000 live births. Note that the added comparison data should always be in the probability scale. |
| ... | unused. |

Examples

```
## Not run:
years <- levels(DemoData[[1]]$time)

# obtain direct estimates
data <- getDirectList(births = DemoData,
  years = years,
  regionVar = "region", timeVar = "time",
  clusterVar = "~clustid+id",
  ageVar = "age", weightsVar = "weights",
  geo.recode = NULL)
# obtain direct estimates
data_multi <- getDirectList(births = DemoData, years = years,
  regionVar = "region", timeVar = "time", clusterVar = "~clustid+id",
  ageVar = "age", weightsVar = "weights", geo.recode = NULL)
data <- aggregateSurvey(data_multi)

fit2 <- fitINLA(data = data, geo = geo, Amat = mat,
  year_label = years.all, year_range = c(1985, 2019),
  rw = 2, is.yearly=TRUE, m = 5, type.st = 4)
out2 <- getSmoothed(fit2, Amat = mat)

plot(out2, is.yearly=TRUE, is.subnational=TRUE)

hatchPlot(data = subset(out2, is.yearly==FALSE), geo = geo,
  variables=c("years"), values = c("median"),
  by.data = "region", by.geo = "REGNAME",
  lower = "lower", upper = "upper", is.long=TRUE)

## End(Not run)
```

| | |
|---------|--|
| iid.new | <i>New random IID models for m-year to period random effects</i> |
|---------|--|

Description

New random IID models for m-year to period random effects

Usage

```
iid.new(cmd = c("graph", "Q", "mu", "initial", "log.norm.const",  
"log.prior", "quit"), theta = NULL)
```

Arguments

| | |
|-------|--------------------------|
| cmd | list of model components |
| theta | log precision |

| | |
|------------|--|
| iid.new.pc | <i>New random IID models for m-year to period random effects</i> |
|------------|--|

Description

New random IID models for m-year to period random effects

Usage

```
iid.new.pc(cmd = c("graph", "Q", "mu", "initial", "log.norm.const",  
"log.prior", "quit"), theta = NULL)
```

Arguments

| | |
|-------|--------------------------|
| cmd | list of model components |
| theta | log precision |

| | |
|---------|---|
| KenData | <i>Auxiliary data for Kenya 2014 DHS.</i> |
|---------|---|

Description

The list contains several data frames.

Usage

```
data(KenData)
```

Format

An object of class `list` of length 4.

Details

- HIV2014, a data frame with three columns: time (in five year periods), region (8 Admin-1 region groups), and the estimated bias of the reported U5MR due to HIV for each 5 year period from 1990-1994 to 2010-2014. The bias is represented as the ratio of the reported U5MR to the true U5MR.
- HIV2014.yearly, a data frame with three columns: time (in five year periods), region (8 Admin-1 region groups), and the estimated bias of the reported U5MR due to HIV for each year from 1980 to 2014. The bias is represented as the ratio of the reported U5MR to the true U5MR.
- IGME2019. Yearly Estimates of national under-5 child mortality in Kenya from the 2019 UN-IGME estimates.
- UrbanProp. Proportion of urban population by county and total population by county. Source: 2009 Kenya Population and Housing Census, and Table A2 of Kenya 2014 DHS report.

References

Neff Walker, Kenneth Hill, and Fengmin Zhao (2012) *Child mortality estimation: methods used to adjust for bias due to aids in estimating trends in under-five mortality.*, *PLoS Medicine*, 9(8):e1001298.

| | |
|------------|---------------------------|
| KingCounty | <i>Map of King County</i> |
|------------|---------------------------|

Description

Shapefiles are King County in the Washington States.

Usage

```
KingCounty
```

Format

An object of class SpatialPolygonsDataFrame with 48 rows and 9 columns.

| | |
|-------|-----------------------------|
| logit | <i>Logit transformation</i> |
|-------|-----------------------------|

Description

Logit transformation

Usage

```
logit(x)
```

Arguments

| | |
|---|------|
| x | data |
|---|------|

Value

logit of x

Examples

```
x <- .5
logit(x)
```

| | |
|---------|------------------------|
| mapPlot | <i>Makes map plot.</i> |
|---------|------------------------|

Description

This function visualizes the map with different variables. The input data frame can be either the long or wide format.

Usage

```
mapPlot(data, variables, values = NULL, labels = NULL, geo, by.data,
  by.geo, is.long = FALSE, size = 0.5, removetab = FALSE,
  border = "gray20", ncol = NULL, ylim = NULL, legend.label = NULL,
  per1000 = FALSE, clean = TRUE)
```

Arguments

| | |
|--------------|--|
| data | a data frame with variables to be plotted |
| variables | vector of variables to be plotted. If long format of data is used, only one variable can be selected |
| values | the column corresponding to the values to be plotted, only used when long format of data is used |
| labels | vector of labels to use for each variable, only used when wide format of data is used |
| geo | geo output from <code>read_shape</code> |
| by.data | column name specifying region names in the data |
| by.geo | variable name specifying region names in the data |
| is.long | logical indicator of whether the data is in the long format, default to FALSE |
| size | size of the border |
| removetab | logical indicator to not show the tab label, only applicable when only one tab is present. |
| border | color of the border |
| ncol | number of columns for the output tabs |
| ylim | range of the values to be plotted. |
| legend.label | Label for the color legend. |
| per1000 | logical indicator to plot mortality rates as rates per 1,000 live births. Note that the added comparison data should always be in the probability scale. |
| clean | remove all coordinates for a cleaner layout, default to TRUE. importFrom sp proj4string |

Examples

```
## Not run:
data(DemoMap)
# Plotting data in the long format
dat <- data.frame(region = rep(c("central", "eastern", "northern", "western"), 3),
  year = rep(c(1980, 1990, 2000), each = 4),
  values = stats::rnorm(12))
utils::head(dat)
mapPlot(dat, variables = "year", values = "values",
  by.data = "region", geo = DemoMap$geo,
  by.geo = "NAME_final", is.long = TRUE)
dat <- data.frame(region = c("central", "eastern", "northern", "western"),
  Year1 = stats::rnorm(4), Year2 = stats::rnorm(4),
  Year3 = stats::rnorm(4))
utils::head(dat)
mapPlot(dat, variables = c("Year1", "Year2", "Year3"),
  labels = c(1980, 1990, 2000),
  by.data = "region", geo = DemoMap$geo,
  by.geo = "NAME_final", is.long = FALSE)
```

```
## End(Not run)
```

mapPoints

Map GPS points to polygon regions

Description

Map GPS points to polygon regions

Usage

```
mapPoints(data, geo, long, lat, names)
```

Arguments

| | |
|-------|---|
| data | point data with two columns of GPS locations. |
| geo | SpatialPolygonsDataFrame of the map |
| long | column name for longitudinal coordinate in the data |
| lat | column name for latitude coordinate in the data |
| names | character vector of region ids to be added to the neighbours list |

Value

Spatial djacency matrix.

Examples

```
data(DemoMap)
dat <- data.frame(ID = c(1,2,3), lon = c(32.2, 33.7, 33), lat = c(0.1, 0.9, 2.8))
dat2 <- mapPoints(dat, DemoMap$geo, long = "lon", lat = "lat", names = "REGNAME")
dat2
```

plot.SUMMERproj *Plot projection output.*

Description

Plot projection output.

Usage

```
## S3 method for class 'SUMMERproj'
plot(x, year_label = c("85-89", "90-94", "95-99",
  "00-04", "05-09", "10-14", "15-19"), year_med = c(1987, 1992, 1997,
  2002, 2007, 2012, 2017), is.subnational = TRUE, proj_year = 2015,
  data.add = NULL, option.add = list(point = NULL, lower = NULL, upper
  = NULL, by = NULL), color.add = "black", label.add = NULL,
  dodge.width = 1, plot.CI = NULL, per1000 = FALSE,
  color.CI = "black", alpha.CI = 0.3, ...)
```

Arguments

| | |
|----------------|--|
| x | output from getSmoothed |
| year_label | labels for the periods |
| year_med | labels for the middle years in each period |
| is.subnational | logical indicator of whether the data contains subnational estimates |
| proj_year | the first year where projections are made, i.e., where no data are available. |
| data.add | data frame for the Comparisons data points to add to the graph. This can be, for example, the raw direct estimates. This data frame is merged to the projections by column 'region' and 'years'. Except for these two columns, this dataset should not have Comparisons columns with names overlapping the getSmoothed output. |
| option.add | list of options specifying the variable names for the points to plot, lower and upper bounds, and the grouping variable. This is intended to be used to add Comparisons estimates on the same plot as the smoothed estimates. See examples for details. |
| color.add | the color of the Comparisons data points to plot. |
| label.add | the label of the Comparisons data points in the legend. |
| dodge.width | the amount to add to data points at the same year to avoid overlap. Default to be 1. |
| plot.CI | logical indicator of whether to plot the error bars. |
| per1000 | logical indicator to plot mortality rates as rates per 1,000 live births. Note that the added comparison data should always be in the probability scale. |
| color.CI | the color of the error bars of the credible interval. |
| alpha.CI | the alpha (transparency) of the error bars of the credible interval. |
| ... | optional arguments, see details |

Details

Note that arguments after ... must match exactly.

- year_labelstring of year labels, defaults to c("85-89", "90-94", "95-99", "00-04", "05-09", "10-14", "15-19")
- proj_yearprojection year as numeric, defaults to 2015
- year_med median of year intervals, defaults to c(1987, 1992, 1997, 2002, 2007, 2012, 2017)
- is_yearlyindicator for yearly model, defaults to TRUE
- is_subnationalindicator for subnational model, defaults to TRUE

See Also

[getSmoothed](#)

Examples

```
## Not run:
years <- levels(DemoData[[1]]$time)

# obtain direct estimates
data <- getDirectList(births = DemoData,
  years = years,
  regionVar = "region", timeVar = "time",
  clusterVar = "~clustid+id",
  ageVar = "age", weightsVar = "weights",
  geo.recode = NULL)
# obtain direct estimates
data_multi <- getDirectList(births = DemoData, years = years,
  regionVar = "region", timeVar = "time", clusterVar = "~clustid+id",
  ageVar = "age", weightsVar = "weights", geo.recode = NULL)
data <- aggregateSurvey(data_multi)

# national model
years.all <- c(years, "15-19")
fit1 <- fitINLA(data = data, geo = NULL, Amat = NULL,
  year_label = years.all, year_range = c(1985, 2019),
  rw = 2, is_yearly=FALSE, m = 5)
out1 <- getSmoothed(fit1)
plot(out1, is_subnational=FALSE)

# subnational model
fit2 <- fitINLA(data = data, geo = geo, Amat = mat,
  year_label = years.all, year_range = c(1985, 2019),
  rw = 2, is_yearly=TRUE, m = 5, type.st = 4)
out2 <- getSmoothed(fit2, Amat = mat)
plot(out2, is_yearly=TRUE, is_subnational=TRUE)

## End(Not run)
```

| | |
|------------|--------------------------------------|
| read_shape | <i>Function to read shape files.</i> |
|------------|--------------------------------------|

Description

Function to read shape files.

Usage

```
read_shape(filepath, regionnames, data = NULL)
```

Arguments

| | |
|-------------|---|
| filepath | file path for .shp files |
| regionnames | vector of strings of final region names |
| data | optional country summary data, for checking |

Value

A list including shape files and the adjacency matrix.

Examples

```
## Not run:  
my_region_names <- c("central", "eastern", "northern", "western")  
my_fp <- "myExampleFilepath/sdr_subnational_boundaries.shp"  
my_map <- read_shape(filepath = my_fp, regionnames = my_region_names)  
  
## End(Not run)
```

| | |
|-----|---|
| rst | <i>Simulate spatial and temporal random effects</i> |
|-----|---|

Description

This function simulates spatial and temporal random effects with mean zero. The method is described in Algorithm 3.1 of Rue & Held 2015.

Usage

```
rst(n = 1, type = c("s", "t", "st")[1], type.s = "ICAR",  
    type.t = c("RW1", "RW2")[2], Amat = NULL, n.t = NULL,  
    scale.model = TRUE)
```

Arguments

| | |
|--------------------------|--|
| <code>n</code> | sample size |
| <code>type</code> | type of random effects: temporal (t), spatial (s), or spatial-temporal (st) |
| <code>type.s</code> | type of spatial random effect, currently only ICAR is available |
| <code>type.t</code> | type of temporal random effect, currently only RW1 and RW2 are available |
| <code>Amat</code> | adjacency matrix for the spatial regions |
| <code>n.t</code> | number of time points for the temporal random effect |
| <code>scale.model</code> | logical indicator of whether to scale the random effects to have unit generalized variance. See Sørbye 2013 for more details |

Value

a matrix (for spatial or temporal) or a three-dimensional array (for spatial-temporal) of the random effects.

References

- Rue, H., & Held, L. (2005). *Gaussian Markov random fields: theory and applications*. CRC press.
- Sørbye, S. H. (2013). *Tutorial: Scaling IGMRF-models in R-INLA*. Department of Mathematics and Statistics, University of Tromsø.

Examples

```
## Not run:
data(DemoMap)
## Spatial random effects
out <- rst(n=10000, type = "s", Amat = DemoMap$Amat)
# To verify the mean under the conditional specification
mean(out[,1] - apply(out[,c(2,3,4)], 1, mean))
mean(out[,2] - apply(out[,c(1,3)], 1, mean))
mean(out[,3] - apply(out[,c(1,2,4)], 1, mean))
mean(out[,4] - apply(out[,c(1,3)], 1, mean))

## Temporal random effects (RW1)
out <- rst(n=1, type = "t", type.t = "RW1", n.t = 200, scale.model = FALSE)
par(mfrow = c(1,2))
plot(1:dim(out)[2], out, col = 1, type = "l", xlab = "Time", ylab = "Random effects")
# verify the first order difference is normally distributed
first_diff <- diff(as.numeric(out[1,]))
qqnorm(first_diff )
abline(c(0,1))

## Temporal random effects (RW2)
out <- rst(n=1, type = "t", type.t = "RW2", n.t = 200, scale.model = FALSE)
par(mfrow = c(1,2))
plot(1:dim(out)[2], out, col = 1, type = "l", xlab = "Time", ylab = "Random effects")
# verify the second order difference is normally distributed
first_diff <- diff(as.numeric(out[1,]))
second_diff <- diff(first_diff)
```

```

qqnorm(second_diff)
abline(c(0,1))

## Spacial-temporal random effects
out <- rst(n=1, type = "st", type.t = "RW1", Amat = DemoMap$Amat, n.t = 50)
dimnames(out)
par(mfrow = c(1,1))
plot(1:dim(out)[3], out[1,1,], col = 1,
     type = "l", ylim = range(out), xlab = "Time", ylab = "Random effects")
for(i in 2:4) lines(1:dim(out)[3], out[1,i,], col = i)
legend("bottomright", colnames(DemoMap$Amat), col = c(1:4), lty = rep(1,4))

## End(Not run)

```

rw.new

New random walk 1 and 2 models for m-year to period random effects

Description

New random walk 1 and 2 models for m-year to period random effects

Usage

```
rw.new(cmd = c("graph", "Q", "mu", "initial", "log.norm.const",
              "log.prior", "quit"), theta = NULL)
```

Arguments

| | |
|-------|--------------------------|
| cmd | list of model components |
| theta | log precision |

rw.new.pc

New random walk 1 and 2 models for m-year to period random effects

Description

New random walk 1 and 2 models for m-year to period random effects

Usage

```
rw.new.pc(cmd = c("graph", "Q", "mu", "initial", "log.norm.const",
                 "log.prior", "quit"), theta = NULL)
```

Arguments

| | |
|-------|--------------------------|
| cmd | list of model components |
| theta | log precision |

`simhyper`*Function to simulate hyperpriors from an adjacency matrix.*

Description

Function to simulate hyperpriors from an adjacency matrix.

Usage

```
simhyper(R = 2, nsamp = 1e+05, nsamp.check = 5000, Amat,  
         nperiod = 6, only.iid = TRUE)
```

Arguments

| | |
|--------------------------|--|
| <code>R</code> | Desired prior odds ratio. Default to 2, i.e., a 95% prior interval for the residual odds ratios lies in the interval (R, 1/R). |
| <code>nsamp</code> | Sample to simulate for scaling factor |
| <code>nsamp.check</code> | Sample to simulate for checking range |
| <code>Amat</code> | Adjacency matrix of the areas in the data. |
| <code>nperiod</code> | numerical value of how many time periods in the data |
| <code>only.iid</code> | Indicator for whether or not only IID hyperpriors are simulated |

References

Wakefield, J. Multi-level modelling, the ecologic fallacy, and hybrid study designs. *International Journal of Epidemiology*, 2009, vol. 38 (pg. 330-336).

Examples

```
## Not run:  
data(DemoMap)  
mat <- DemoMap$Amat  
priors <- simhyper(R = 2, nsamp = 1e+05, nsamp.check = 5000, Amat = mat)  
  
## End(Not run)
```

| | |
|--------|---|
| st.new | <i>New Type I to IV space time interaction models for m-year to period random effects</i> |
|--------|---|

Description

New Type I to IV space time interaction models for m-year to period random effects

Usage

```
st.new(cmd = c("graph", "Q", "mu", "initial", "log.norm.const",
              "log.prior", "quit"), theta = NULL)
```

Arguments

| | |
|-------|--------------------------|
| cmd | list of model components |
| theta | log precision |

| | |
|-----------|---|
| st.new.pc | <i>New Type I to IV space time interaction models for m-year to period random effects</i> |
|-----------|---|

Description

New Type I to IV space time interaction models for m-year to period random effects

Usage

```
st.new.pc(cmd = c("graph", "Q", "mu", "initial", "log.norm.const",
                 "log.prior", "quit"), theta = NULL)
```

Arguments

| | |
|-------|--------------------------|
| cmd | list of model components |
| theta | log precision |

Index

*Topic **datasets**

- BRFSS, [4](#)
- DemoData, [5](#)
- DemoData2, [6](#)
- DemoMap, [6](#)
- DemoMap2, [7](#)
- KenData, [27](#)
- KingCounty, [27](#)

aggregateSurvey, [3](#)

BRFSS, [4](#)

ChangeRegion, [4](#), [20](#), [22](#)

DemoData, [5](#)

DemoData2, [6](#)

DemoMap, [6](#)

DemoMap2, [7](#)

expit, [7](#)

fitGeneric, [8](#)

fitINLA, [9](#), [10](#), [19](#), [23](#)

fitINLA2, [12](#)

getAdjusted, [14](#)

getAmat, [15](#)

getBirths, [16](#), [20](#), [21](#)

getCounts, [18](#)

getDiag, [19](#)

getDirect, [11](#), [14](#), [20](#), [22](#)

getDirectList, [3](#), [9](#), [21](#), [21](#)

getSmoothed, [22](#), [31](#), [32](#)

hatchPlot, [24](#)

iid.new, [26](#)

iid.new.pc, [26](#)

KenData, [27](#)

KingCounty, [27](#)

logit, [28](#)

mapPlot, [28](#)

mapPoints, [30](#)

plot.SUMMERproj, [23](#), [31](#)

read_shape, [24](#), [29](#), [33](#)

rst, [33](#)

rw.new, [35](#)

rw.new.pc, [35](#)

simhyper, [10](#), [13](#), [36](#)

st.new, [37](#)

st.new.pc, [37](#)

SUMMER (SUMMER-package), [3](#)

SUMMER-package, [3](#)