Package: surveyPrev (via r-universe)

September 10, 2024

Type Package

Title Mapping the Prevalence of Binary Indicators using Survey Data in Small Areas

Version 1.0.0

Description Provides a pipeline to perform small area estimation and prevalence mapping of binary indicators using health and demographic survey data, described in Fuglstad et al. (2022) <doi:10.48550/arXiv.2110.09576> and Wakefield et al. (2020) <doi:10.1111/insr.12400>.

URL `https://github.com/richardli/surveyPrev`

BugReports `https://github.com/richardli/surveyPrev/issues`

Depends R (>= 3.5)

License GPL (>= 2)

Imports survey, stats, ggplot2, ggridges, rdhs, SUMMER, dplyr, labelled, sjlabelled, naniar, raster, sp, spdep, stringr, tidyverse, data.table, sf, matrixStats, rlang

Encoding UTF-8

LazyData true

RoxygenNote 7.3.1

Additional_repositories `https://inla.r-inla-download.org/R/testing/`

Suggests INLA, knitr, rmarkdown, R.rsp, kableExtra, geodata, patchwork, tidyr

VignetteBuilder R.rsp, knitr

NeedsCompilation no

Config/build/clean-inst-doc FALSE

Repository https://richardli.r-universe.dev

RemoteUrl https://github.com/richardli/surveyprev

RemoteRef HEAD

RemoteSha e5a603e430ff4fda992c4aa55612cb83e9681296

Contents

adminInfo *Get admin information*

Description

This function get admin information including name, character, population and unban/rural proportion.

Usage

```
adminInfo(
  poly.adm,
  by.adm,
  admin,
  by.adm.upper = NULL,
  agg.pop = NULL,
  proportion = NULL
)
```

Arguments

`poly.adm`	spatial polygons dataframe for Admin levels such as Admin 1 or Admin 2. This object can be either an sp::SpatialPolygonsDataFrame object or an sf object.
`by.adm`	the column name of column for Admin names for desired output Admin level, can be such as "NAME_1" or "NAME_2".
`admin`	desired admin level for the output, can be 1 or 2.
`by.adm.upper`	the column name of column for Admin names for upper level of your desired output Admin level when admin=2, can be "NAME_1" when by.adm="NAME_2".
`agg.pop`	data frame of aggregated poplulation from aggPopulation function. It should have two columns: "admin2.name.full" and "population".
`proportion`	data frame of urban/rural proportions. For admin1, is should have two columns: "admin1.name" and "urban". For admin2, it should have three columns: "admin1.name", "admin2.name", and "urban", in order to avoid issues merging datasets with duplicated admin2 names.

Value

This function returns the 1. dataframe that contains admin 1 and admin 2 information and coordinates for each cluster and 2. Adjacency matrix.

Author(s)

Qianyu Dong

Examples

```
# For sp::SpatialPolygonsDataFrame object
data(ZambiaAdm1)
class(ZambiaAdm1)
info <- adminInfo(poly.adm=ZambiaAdm1, admin = 1, by.adm="NAME_1")
data(ZambiaAdm2)
class(ZambiaAdm2)
info2 <- adminInfo(poly.adm=ZambiaAdm2, admin = 2,by.adm="NAME_2",by.adm.upper="NAME_1")

# For sf object
geo.sf <- sf::st_as_sf(ZambiaAdm1)
info <- adminInfo(poly.adm=geo.sf, admin = 1,by.adm="NAME_1")

# To include the population information
data(ZambiaPopWomen)
info <- adminInfo(poly.adm = ZambiaAdm1,
                  admin = 1,by.adm="NAME_1",
                  agg.pop = ZambiaPopWomen$admin1_pop,
                  proportion = ZambiaPopWomen$admin1_urban )
```

aggPopulation *Get population information*

Description

This function aggregate population to particular admin levels

Usage

```
aggPopulation(tiff, fact = 10, poly.adm, by.adm, by.adm.upper = NULL)
```

Arguments

`tiff`	spatial raster of population estimates.
`fact`	factor to aggregate pixels. Default to be 10, i.e., the population estimates will be saved on 1km by 1km grids if the input is 100m by 100m tiff. Larger values of aggregation factor improves the computation speed, but can introduce more errors when the regions defined by the polygon are small in size.
`poly.adm`	spatial polygons dataframe.
`by.adm`	the column name of column for Admin names for desired output Admin level, can be such as "NAME_1" or "NAME_2". Duplicated admin names are handled internally when by.adm.upper is specified
`by.adm.upper`	the column name of column for Admin names for upper level of your desired output Admin level when admin=2, can be "NAME_1" when by.adm="NAME_2".

Value

This function returns the dataset that contain district name and population for given tiff files and polygons of admin level

Author(s)

Qianyu Dong

Examples

```
## Not run:
library(raster)

# Download and find total population in age group 0 to 12 months
pre <- "https://data.worldpop.org/GIS/AgeSex_structures/"
f <- paste0(pre, "Global_2000_2020/2018/ZMB/zmb_f_0_2018.tif")
m <- paste0(pre, "Global_2000_2020/2018/ZMB/zmb_m_0_2018.tif")
pop_f_0 <- raster(f)
pop_m_0 <- raster(m)

pop_raster <- pop_f_0 + pop_m_0

# admin1 population
agg.pop1 <- aggPopulation(
  tiff = pop_raster,
  poly.adm = ZambiaAdm1,
  by.adm = "NAME_1")


# admin2 population
agg.pop2 <- aggPopulation(
  tiff = ZambiaPopWomen_raster,
  poly.adm = ZambiaAdm2,
  by.adm = "NAME_2",
  by.adm.upper="NAME_1")


## End(Not run)
```

aggSurveyWeight *Get survey weight by admin levels*

Description

This function aggregate survey weight to particular admin levels

Usage

```
aggSurveyWeight(
  data,
  cluster.info,
  admin,
  poly.adm = NULL,
  by.adm = NULL,
  by.adm.upper = NULL
)
```

Arguments

`data` dataframe that contains the indicator of interests, output of getDHSindicator function

`cluster.info` list that contains admin 1 and admin 2 information and coordinates for each cluster, output of clusterinfo function

`admin` desired admin level for aggregation

`poly.adm` spatial polygons dataframe

`by.adm` the column name of column for Admin names for desired output Admin level, can be such as "NAME_1" or "NAME_2".

`by.adm.upper` the column name of column for Admin names for upper level of your desired output Admin level when admin=2, can be "NAME_1" when by.adm="NAME_2".

Value

This function returns the dataset that contain admin name and survey weight.

Author(s)

Qianyu Dong

Examples

```
## Not run:

# admin1 population

year <- 2018
country <- "Zambia"
indicator="nmr"

geo <- getDHSgeo(country = country, year = year)
dhsData <- getDHSdata(country = country, indicator=indicator, year = year)
data<- getDHSindicator(dhsData, indicator = indicator)

poly.adm1=ZambiaAdm1
poly.adm2=ZambiaAdm2

cluster.info<-clusterInfo(geo=geo, poly.adm1=poly.adm1, poly.adm2=poly.adm2,
by.adm1 = "NAME_1",by.adm2 = "NAME_2")

agg.survey1<-aggSurveyWeight(data=data,cluster.info=cluster.info,admin=1)
agg.survey2<-aggSurveyWeight(data=data,cluster.info=cluster.info,admin=2,
                             poly.adm = poly.adm2,  by.adm="NAME_2",
                             by.adm.upper ="NAME_1")

## End(Not run)
```

ch_allvac_either *CH_VACS_C_BAS Children with all 8 basic vaccinations (age 12-23) "All basic vaccinations according to either source"*

Description

CH_VACS_C_BAS Children with all 8 basic vaccinations (age 12-23) "All basic vaccinations according to either source"

Usage

```
ch_allvac_either(Rdata)
```

Arguments

Rdata	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "CH_VACS_C_BAS",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::ch_allvac_either)

## End(Not run)
```

ch_diar_ors_rhf *CH_DIAT_C_ORT KR Diarrhea treatment (Children under five with diarrhea treated with either ORS or RHF)*

Description

CH_DIAT_C_ORT KR Diarrhea treatment (Children under five with diarrhea treated with either ORS or RHF)

Usage

```
ch_diar_ors_rhf(Rdata)
```

Arguments

Rdata	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "CH_DIAT_C_ORT",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::ch_diar_ors_rhf)

## End(Not run)
```

ch_meas_either *CH_VACC_C_MSL MCV: Measles Measles vaccination received Percentage of children (age 12-23) ch_meas_either CH_VAC.do KR "Measles vaccination according to either source"*

Description

CH_VACC_C_MSL MCV: Measles Measles vaccination received Percentage of children (age 12-23) ch_meas_either CH_VAC.do KR "Measles vaccination according to either source"

Usage

```
ch_meas_either(Rdata)
```

Arguments

`Rdata`	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "CH_VACC_C_MSL",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                        FUN = surveyPrev::ch_meas_either)

## End(Not run)
```

`ch_novac_either` *CH_VACS_C_NON KR Children with no vaccinations (age 12-23)*

Description

CH_VACS_C_NON KR Children with no vaccinations (age 12-23)

Usage

```
ch_novac_either(Rdata)
```

Arguments

`Rdata`	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "CH_VACS_C_NON",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::ch_novac_either)

## End(Not run)
```

ch_pent1_either *CH_VACC_C_DP1 KR Percentage of children (age 12-23) Pentavalent 1rd dose vaccination according to either source"*

Description

CH_VACC_C_DP1 KR Percentage of children (age 12-23) Pentavalent 1rd dose vaccination according to either source"

Usage

```
ch_pent1_either(Rdata)
```

Arguments

Rdata	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "CH_VACC_C_DP1",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::ch_pent1_either)

## End(Not run)
```

ch_pent3_either *CH_VACC_C_DP3 DPT3 KR Percentage of children (age 12-23) Pentavalent 3rd dose vaccination according to either source"*

Description

CH_VACC_C_DP3 DPT3 KR Percentage of children (age 12-23) Pentavalent 3rd dose vaccination according to either source"

Usage

```
ch_pent3_either(Rdata)
```

Arguments

`Rdata`	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "CH_VACC_C_DP3",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                        FUN = surveyPrev::ch_pent3_either)

## End(Not run)
```

clusterInfo *Get cluster information*

Description

This function add admin 1 and admin2 information to a paticular DHS survey.

Usage

```
clusterInfo(geo, poly.adm1, poly.adm2, by.adm1 = "NAME_1", by.adm2 = "NAME_2")
```

Arguments

`geo` spatial point dataframe

`poly.adm1` spatial polygons dataframe for admin 1

`poly.adm2` spatial polygons dataframe for admin 2 or other lower admin level.

`by.adm1` the column name of column for Admin names for admin 1

`by.adm2` the column name of column for Admin names for admin 2 or other lower admin level.

Value

This function returns the dataset that contains admin 1 and admin 2 information and coordinates for each cluster.

Author(s)

Qianyu Dong

Examples

```
## Not run:
geo <- getDHSgeo(country = "Zambia", year = 2018)
data(ZambiaAdm1)
data(ZambiaAdm2)
cluster.info <- clusterInfo(geo = geo,
                            poly.adm1 = ZambiaAdm1,
                            poly.adm2 = ZambiaAdm2)

## End(Not run)
```

clusterModel *Calculate cluster model estimates using beta binomial model*

Description

This function calculate smoothed direct estimates at given admin level.

Usage

```
clusterModel(
  data,
  cluster.info,
  admin.info,
  X = NULL,
  admin,
  CI = 0.95,
  model = c("bym2", "iid"),
  stratification = FALSE,
  aggregation = FALSE,
  overdisp.mean = 0,
  overdisp.prec = 0.4,
  pc.u = 1,
  pc.alpha = 0.01,
  pc.u.phi = 0.5,
  pc.alpha.phi = 2/3
)
```

Arguments

`data` dataframe that contains the indicator of interests(column name is value), output of getDHSindicator function

`cluster.info` dataframe that contains admin 1 and admin 2 information and coordinates for each cluster.

`admin.info` dataframe that contains population and urban/rural proportion at specific admin level

`X` dataframe that contains areal covariates, the first column should be the same admin name as in admin.info$data.

`admin` admin level for the model

`CI` Credible interval to be used. Default to 0.95.

`model` smoothing model used in the random effect. Options are independent ("iid") or spatial ("bym2").

`stratification` whether or not to include urban/rural stratum.

`aggregation` whether or not report aggregation results.

`overdisp.mean` prior mean for logit(d), where d is the intracluster correlation.

`overdisp.prec` prior precision for logit(d), where d is the intracluster correlation.

`pc.u` pc prior u for iid or bym2 precision.

`pc.alpha` pc prior alpha for iid or bym2 precision.

`pc.u.phi` pc prior u for bym2 mixing paramete.

`pc.alpha.phi` pc prior u for bym2 mixing paramete.

Value

This function returns the dataset that contain district name and population for given tiff files and polygons of admin level,

Author(s)

Qianyu Dong

Examples

```
## Not run:
geo <- getDHSgeo(country = "Zambia", year = 2018)
data(ZambiaAdm1)
data(ZambiaAdm2)
data(ZambiaPopWomen)
cluster.info <- clusterInfo(geo = geo,
                            poly.adm1 = ZambiaAdm1,
                            poly.adm2 = ZambiaAdm2)

dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "ancvisit4+",
                                 year = 2018)

data <- getDHSindicator(dhsData, indicator = "ancvisit4")
admin.info1 <- adminInfo(poly.adm = ZambiaAdm1,
                        admin = 1,
                        agg.pop =ZambiaPopWomen$admin1_pop,
                        proportion = ZambiaPopWomen$admin1_urban)
cl_res_ad1 <- clusterModel(data=data,
                  cluster.info = cluster.info,
                  admin.info = admin.info1,
                  stratification = FALSE,
                  model = "bym2",
                  admin = 1,
                  aggregation = TRUE,
                  CI = 0.95)
cl_res_ad1$res.admin1

# compare with the DHS direct estimates
dhs_table <- get_api_table(country = "ZM",
                           survey = "ZM2018DHS",
                           indicator = "RH_ANCN_W_N4P",
                           simplify = TRUE)
subset(dhs_table, ByVariableLabel == "Five years preceding the survey")


## End(Not run)
```

directEST *Calculate direct estimates*

Description

This function calculate direct estimates at given admin level.

Usage

```
directEST(
  data,
  cluster.info,
  admin,
  strata = "all",
  CI = 0.95,
  weight = c("population", "survey")[1],
  admin.info = NULL,
  aggregation = FALSE,
  alt.strata = NULL,
  ...
)
```

Arguments

`data` dataframe that contains the indicator of interests, output of getDHSindicator function

`cluster.info` list contains data and wrong.points. data contains admin 1 and admin 2 information and coordinates for each cluster. wrong.points. contains cluster id for cluster without coordinates or admin 1 information. Output of getDHSindicator function

`admin` admin level for the model.

`strata` use only urban or rural data, only for national level model

`CI` Credible interval to be used. Default to 0.95.

`weight` the weight used for aggregating result, "population" or "survey"

`admin.info` list contains data and mat, data contains population and urban/rural proportion at specific admin level and mat is the adjacency matrix, output of adminInfo function

`aggregation` whether or not report aggregation results.

`alt.strata` the variable name in the data frame that correspond to the stratification variable. Most of the DHS surveys are stratified by admin 1 area crossed with urban/rural, which is the default stratification variable created by the function (when `alt.strata = NULL`). When a different set of strata is used. The stratification variable should be included in the data and `alt.strata` should be set to the column name.

`...` Additional arguments passed on to the ‘smoothSurvey‘ function

Value

This function returns the dataset that contain district name and population for given tiff files and polygons of admin level,

Author(s)

Qianyu Dong

Examples

```
## Not run:

##
## Direct estimation of ANC visit 4+ proportion
##

geo <- getDHSgeo(country = "Zambia", year = 2018)
data(ZambiaAdm1)
data(ZambiaAdm2)
data(ZambiaPopWomen)
cluster.info<-clusterInfo(geo=geo, poly.adm1=ZambiaAdm1, poly.adm2=ZambiaAdm2,
by.adm1 = "NAME_1",by.adm2 = "NAME_2")
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "ancvisit4+",
                                 year = 2018)

data <- getDHSindicator(dhsData, indicator = "ancvisit4+")
res_ad1 <- directEST(data = data,
                  cluster.info = cluster.info,
                  admin = 1,
                  aggregation = FALSE)
res_ad1
# compare with the DHS direct estimates
dhs_table <- get_api_table(country = "ZM",
                           survey = "ZM2018DHS",
                           indicator = "RH_ANCN_W_N4P",
                           simplify = TRUE)
subset(dhs_table, ByVariableLabel == "Five years preceding the survey")

##
## Changing customized stratification variable
##

data_alt <- data
# Assuming the stratification is done with only admin1 area
# and not stratified by urban and rural
# Note that this is not the correct stratification, but we use
#  this as an illustration to create user-specified strata variable
data_alt$new_strata <- data_alt$v024
res_ad1_wrong <- directEST(data = data_alt,
                  cluster.info = cluster.info,
                  admin = 1,
                  aggregation = FALSE,
                  alt.strata = "new_strata")
res_ad1_wrong

## End(Not run)
```

exceedPlot *Plot exceedance probability of model results*

Description

This function return a map of exceedance probability for given model results.

Usage

```
exceedPlot(
  x,
  exceed = TRUE,
  direction = 1,
  threshold = NA,
  geo = geo,
  by.geo = NULL,
  ylim = NULL,
  ...
)
```

Arguments

`x`	a model result using surveyPrev of class "fhModel" or "clusterModel"
`exceed`	direction of the comparison The default is exceed = TRUE, which correspond to probability of prevalence larger than the threshold. If exceed = FALSE, the plot computes probability smaller than the threshold.
`direction`	Direction of the color scheme. It can be either 1 (smaller values are darker) or -1 (higher values are darker). Default is set to 1.
`threshold`	the threshold to be used in computing the exceedance probability
`geo`	SpatialPolygonsDataFrame object for the map
`by.geo`	variable name specifying region names in the data
`ylim`	range of the values to be plotted.
`...`	additional arguments passed to SUMMER::mapPlot().

Value

This function returns a map showing probability of prevalence over/under the threshold.

Author(s)

Zehang Li, Qianyu Dong

Examples

```
## Not run:

geo <- getDHSgeo(country = "Zambia", year = 2018)
data(ZambiaAdm1)
data(ZambiaAdm2)
data(ZambiaPopWomen)
cluster.info <- clusterInfo(geo = geo,
                            poly.adm1 = ZambiaAdm1,
                            poly.adm2 = ZambiaAdm2)

dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "ancvisit4+",
                                 year = 2018)

data <- getDHSindicator(dhsData, indicator = "ancvisit4+")
admin.info2 <- adminInfo(poly.adm = ZambiaAdm2,
                        admin = 2,
         by.adm="NAME_2",
         by.adm.upper = "NAME_1")
cl_res_ad2_unstrat <- clusterModel(data = data,
                  cluster.info = cluster.info,
                  admin.info = admin.info2,
                  stratification = FALSE,
                  model = "bym2",
                  admin = 2,
                  aggregation = TRUE,
                  CI = 0.95)
ZambiaAdm2$admin2.name.full <- paste0(ZambiaAdm2$NAME_1,
              "_",
           ZambiaAdm2$NAME_2)
exceedPlot(cl_res_ad2_unstrat, threshold = 0.5,
     exceed = TRUE, direction = -1,
     geo = ZambiaAdm2, by.geo = "admin2.name.full")
exceedPlot(cl_res_ad2_unstrat, threshold = 0.5,
     exceed = FALSE, direction = -1,
     geo = ZambiaAdm2, by.geo = "admin2.name.full")


## End(Not run)
```

fhModel *Calculate smoothed direct estimates*

Description

This function calculate smoothed direct estimates at given admin level.

Usage

```
fhModel(
  data,
  cluster.info,
  admin.info = NULL,
  X = NULL,
  admin,
  CI = 0.95,
  model = c("bym2", "iid"),
  aggregation = FALSE,
  alt.strata = NULL,
  ...
)
```

Arguments

`data` dataframe that contains the indicator of interests, output of getDHSindicator function

`cluster.info` list contains data and wrong.points. data contains admin 1 and admin 2 information and coordinates for each cluster. wrong.points. contains cluster id for cluster without coordinates or admin 1 information. Output of getDHSindicator function

`admin.info` list contains data and mat, data contains population and urban/rural proportion at specific admin level and mat is the adjacency matrix, output of adminInfo function

`X` dataframe that contains areal covariates, the first column should be the same admin name as in admin.info$data.

`admin` admin level for the model

`CI` Credible interval to be used. Default to 0.95.

`model` smoothing model used in the random effect. Options are independent ("iid") or spatial ("bym2").

`aggregation` whether or not report aggregation results.

`alt.strata` the variable name in the data frame that correspond to the stratification variable. Most of the DHS surveys are stratified by admin 1 area crossed with urban/rural, which is the default stratification variable created by the function (when `alt.strata = NULL`). When a different set of strata is used. The stratification variable should be included in the data and `alt.strata` should be set to the column name.

`...` Additional arguments passed on to the ‘smoothSurvey‘ function

Value

This function returns the dataset that contain district name and population for given tiff files and polygons of admin level,

Author(s)

Qianyu Dong

Examples

```
## Not run:
geo <- getDHSgeo(country = "Zambia", year = 2018)
data(ZambiaAdm1)
data(ZambiaAdm2)
data(ZambiaPopWomen)
cluster.info <- clusterInfo(geo = geo,
                            poly.adm1 = ZambiaAdm1,
                            poly.adm2 = ZambiaAdm2)

dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "ancvisit4+",
                                 year = 2018)

data <- getDHSindicator(dhsData, indicator = "ancvisit4+")
admin.info1 <- adminInfo(poly.adm = ZambiaAdm1,
                        admin = 1,
                        agg.pop =ZambiaPopWomen$admin1_pop,
                        proportion = ZambiaPopWomen$admin1_urban)
smth_res_ad1 <- fhModel(data,
                       cluster.info = cluster.info,
                       admin.info = admin.info1,
                       admin = 1,
                       model = "bym2",
                       aggregation = F)
smth_res_ad1

## End(Not run)
```

fp_cruse_mod *FP_CUSM_W_MOD IRdata Modern contraceptive prevalence rate (Married women currently using any modern method of contraception)*

Description

FP_CUSM_W_MOD IRdata Modern contraceptive prevalence rate (Married women currently using any modern method of contraception)

Usage

```
fp_cruse_mod(Rdata)
```

Arguments

`Rdata`	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "FP_CUSA_W_MOD",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::fp_cruse_mod)

## End(Not run)
```

`fp_unmet_tot` *FP_NADA_W_UNT #unmet_family IRdata women with an unmet need for family planning for spacing and limiting*

Description

FP_NADA_W_UNT #unmet_family IRdata women with an unmet need for family planning for spacing and limiting

Usage

```
fp_unmet_tot(Rdata)
```

Arguments

`Rdata`	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                  indicator = "FP_NADA_W_UNT",
                                  year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::fp_unmet_tot)

## End(Not run)
```

---

getDHSdata *Download DHS survey data*

Description

This function downloads DHS data for a particular country and survey.

Usage

```
getDHSdata(country, indicator = NULL, Recode = NULL, year)
```

Arguments

`country`	Country name.
`indicator`	Indicator of interests. Current list of supported indicators include: "womananemia", "ancvisit4+", "stunting", "wasting", "DPT3".
`Recode`	Types of dhs Recode
`year`	Year the survey conducted.

Value

This function returns the survey dataset that contains the indicator.

Author(s)

Qianyu Dong

Examples

```
## Not run:
# When indicator is known, download only the relevant file
dhsData <- getDHSdata(country = "Zambia",
                                  indicator = "ancvisit4+",
                                  year = 2018)

# When indicator is NULL or not recognized, download all files
dhsData <- getDHSdata(country = "Zambia",
                                  indicator = NULL,
                                        year = 2018)
names(dhsData)

## End(Not run)
```

getDHSgeo *Download DHS geo data*

Description

This function downloads cluster's coordinate data for country and survey.

Usage

```
getDHSgeo(country, year)
```

Arguments

country	Country name.
year	Year the survey conducted.

Value

The function returns a spatial point dataset with coordinates for each cluster based on the chosen survey and year.

Author(s)

Qianyu Dong

Examples

```
## Not run:
geo <- getDHSgeo(country = "Zambia", year = 2018)

## End(Not run)
```

`getDHSindicator` *Process DHS data*

Description

This function processes DHS data from getDHSdata function.

Usage

```
getDHSindicator(
  Rdata,
  indicator = NULL,
  FUN = NULL,
  nmr.year = 10,
  filter = NULL,
  yesCondition = NULL,
  noCondition = NULL
)
```

Arguments

`Rdata`	Result from getDHSdata function, the raw DHS survry data from get_datasets.
`indicator`	Indicator of interests.
`FUN`	a function to process the DHS data into a binary indicator if not using one of the implemented indicators. See surveyPrev::AN_ANEM_W_ANY for an example function to obtain the indicator for women classified as having any anemia.
`nmr.year`	This is an argument specifically for NMR calculation. It specifies births how many years do we include prior to the date of survey. Default to be 10, i.e., NMR in the last 10 years prior to survey.
`filter`	This arguments specifies how the data should be filtered for creating a customized indicator. It should be a character string or a vector of character strings specifying the expression used to filter the data. See example for details
`yesCondition`	This arguments specifies how to define a yes label, i.e., value = 1, for creating a customized indicator. It should be a character string specifying the expression used to define the outcome value equal to 1. See example for details.
`noCondition`	This arguments specifies how to define a no label, i.e., value = 0, for creating a customized indicator. It should be a character string specifying the expression used to define the outcome value equal to 0. See example for details.

Value

The function returns processed survey data that contains the indicator of interests.

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData1 <- getDHSdata(country = "Zambia",
                                 indicator = "ancvisit4+",
                                 year = 2018)
data1 <- getDHSindicator(dhsData1, indicator = "ancvisit4+")


#------------------------------------------------------------------#
# User-specified function to process the data
# For example see the internal function surveyPrev::AN_ANEM_W_ANY
#------------------------------------------------------------------#
dhsData2 <- getDHSdata(country = "Zambia",
                                 indicator = NULL,
                                 year = 2018)
data2 <- getDHSindicator(dhsData2, indicator = NULL,
                         FUN = surveyPrev::AN_ANEM_W_ANY)
# which should be identical to the following
dhsData3 <- getDHSdata(country = "Zambia",
                                 indicator = "womananemia",
                                 year = 2018)
data3 <- getDHSindicator(dhsData3, indicator = "womananemia")

#------------------------------------------------------------------#
# User-specified filtering condition
# Demonstrating NMR data preparation by specifying how to subset data
#    and specify the outcome variable and its levels
#------------------------------------------------------------------#
Recode <- "Births Recode"
dhsData4 <- getDHSdata(country = "Zambia", indicator = NULL,
                       Recode=Recode,year = "2018")
#
# Here we filter the births to be within the last 10 years
#   this is specified by condition = "v008 - b3 < 120"
# b3 is the date of births in CMC format
# v008 is the date of interview in CMC format
# the difference is the number of months between the two dates
# b7 is the age of death for the child. We consider neonatal deaths where
#   b7 = 0.
# b7 = NA when the child is alive at the time of interview.
data4 <- getDHSindicator(Rdata = dhsData4,
                        indicator = NULL,
                        filter = "v008 - b3 < 120",
                        yesCondition = "b7 == 0",
                        noCondition = "b7 > 0 | is.na(b7)")

# This will return the same dataset as below
data5 <- getDHSindicator(Rdata = dhsData4, indicator = "nmr")

# Notice that filter can have more than one conditions specified by vector
# The following four specifications lead to the same dataset for
#  neonatal deaths in the last 5 years
```

```
data6a <- getDHSindicator(Rdata = dhsData4,
                          indicator = NULL,
                          filter = "v008 - b3 < 120 & v008 - b3 < 60",
                          yesCondition = "b7 == 0",
                          noCondition = "b7 > 0 | is.na(b7)")
data6b <- getDHSindicator(Rdata = dhsData4,
                          indicator = NULL,
                          filter = c("v008 - b3 < 120", "v008 - b3 < 60"),
                          yesCondition = "b7 == 0",
                          noCondition = "b7 > 0 | is.na(b7)")
data6c <- getDHSindicator(Rdata = dhsData4,
                          indicator = NULL,
                          filter = "v008 - b3 < 60",
                          yesCondition = "b7 == 0",
                          noCondition = "b7 > 0 | is.na(b7)")
data7 <- getDHSindicator(Rdata = dhsData4, indicator = "nmr", nmr.year = 5)

## End(Not run)
```

getUR — *Function to threshold population raster to obtain urban/rural fractions by Admin1 and Admin2 areas*

Description

This function computes the urban proportion at a given survey year. It requires two population raster files and urban population fraction by admin 1 area from the census. The census year overall population raster is used to partition the grids into urban and rural pixels, based on the urban population fractions in a given area at the census year. The thresholding process is performed by first sorting the pixels from high to low population density, and find a threshold such that the fraction of population above this threshold matches the urban population fraction from the census. This step defines the urbanicity of each pixel. In the second step, for any given year's raster for a specific (sub-)population (e.g., specific age groups), we aggregate the population in the urban pixels defined in the previous step to compute urban proportion for the (sub-)population, within both admin1 and admin2 regions.

Usage

```
getUR(
  tiff.census,
  tiff.survey,
  prop.census,
  fact = 10,
  poly.adm1,
  poly.adm2,
  varname1,
  varname2
)
```

Arguments

`tiff.census` spatial raster of population estimates at the census year when the sampling frame is based, for the whole population.

`tiff.survey` spatial raster of population estimates at the survey year, for the target population.

`prop.census` a data frame with two columns: ‘admin1‘ column correspond to the admin 1 names in the ‘poly.adm1‘ file. And ‘frac‘ column specifying the proportion of population in each admin 1 area during the census year. See examples for detail.

`fact` factor to aggregate pixels from tiff.survey to tiff.census. For example, if tiff.census is a population raster at 1km by 1km resolution and tiff.survey is a raster at 100m by 100m resolution, then fact should be set to 10. Currently we only support fact > 1. Default is 10.

`poly.adm1` spatial polygons data frame for admin 1

`poly.adm2` spatial polygons data frame for admin 2

`varname1` column name of district name in the admin 1 spatial polygon data frame

`varname2` column name of district name in the admin 2 spatial polygon data frame

Value

a list of two data frames for admin 1 and admin 2 urban ratios

Examples

```
## Not run:
# ------------------------------------------------------------------------#
# Here we consider the example of computing urban/rural fraction for
#  Zambia 2018 DHS for the sub-population of children under 1 years old.
# This survey is based on sampling frame from the 2010 Zambia Census.
# ------------------------------------------------------------------------#
#
# From Table A1 of Zambia 2013-2014 DHS final report, we can obtain the fraction of
#   urban population by Admin 1 areas in the 2010 survey.
# Notice that in the appendix of the 2018 DHS final report,
#   only distribution of household is reported and not population size by urbanicity.
# When the table is not provided in the DHS report, you need to find it from
#   the census website directly.
# Please note that the admin1 column needs to match the admin 1 names in the
#   Admin 1 spatial polygon file exactly.
#   For example, here we change "Northwestern" to "North-Western"

urban.frac <- data.frame(
  admin1 = c('Central', 'Copperbelt', 'Eastern',
       'Luapula', 'Lusaka', 'Muchinga',
     'North-Western', 'Northern', 'Southern','Western'),
  frac = c(0.2513, 0.809, 0.1252,
     0.1963, 0.8456, 0.1714,
     0.2172, 0.1826, 0.2448, 0.1474))
# The corresponding census year population tiff can be found at:
# https://data.worldpop.org/GIS/Population/Global_2000_2020_1km_UNadj/
# The code below downloads the file from the internet directly
# You can also download the file directly and read into R
link1="https://data.worldpop.org/GIS/Population/Global_2000_2020_1km_UNadj/"
file1="2010/ZMB/zmb_ppp_2010_1km_Aggregated_UNadj.tif"
tempfile1 = tempfile()
download.file(paste0(link1, file1), destfile = tempfile1,
        method = "libcurl", mode="wb")
library(raster)
tiff1 <- raster(tempfile1)

# https://hub.worldpop.org/geodata/summary?id=16429
# Here we compute population fractions for 0-1 year old population.
# The from the same link below
link2="https://data.worldpop.org/GIS/AgeSex_structures/Global_2000_2020/"
# The two files are for female and male population respectively,
file2f="2018/ZMB/zmb_f_0_2018.tif"
file2m="2018/ZMB/zmb_f_0_2018.tif"

# Since the two files are very large, we recommend downloading them
#   mannually and then load them into R.
tiff2f <- raster("zmb_f_0_2018.tif")
tiff2m <- raster("zmb_m_0_2018.tif")
tiff2 <- tiff2f + tiff2m

frac <- getUR(tiff.census = tiff1, tiff.survey = tiff2,
     prop.census = urban.frac, fact = 10,
     poly.adm1 = ZambiaAdm1, poly.adm2 = ZambiaAdm2,
     varname1 = "NAME_1", varname2 = "NAME_2")

library(SUMMER)
mapPlot(frac$admin1.ur, geo = ZambiaAdm1,
     by.data = "admin1.name", by.geo = "NAME_1", variable = "urban")
mapPlot(frac$admin2.ur, geo = ZambiaAdm2,
     by.data = "admin2.name", by.geo = "NAME_2", variable = "urban")
# Compare with the proportion of Women 14-49 years old in the built-in data
# These two plots should be similar but not identical
#   since the population is different
mapPlot(ZambiaPopWomen$admin2_urban, geo = ZambiaAdm2,
     by.data = "admin2.name", by.geo = "NAME_2", variable = "urban")


## End(Not run)
```

`get_api_table` *Function to obtain subnational estimates from DHS API*

Description

Function to obtain subnational estimates from DHS API

Usage

```
get_api_table(country, survey, indicator, simplify = TRUE, admin = 1)
```

Arguments

`country` A character string of keys at: https://api.dhsprogram.com/rest/dhs/countries?returnFields=CountryName,

`survey` A character string of keys at: https://api.dhsprogram.com/rest/dhs/surveys?returnFields=SurveyId,Survey

`indicator` A character string of keys at: https://api.dhsprogram.com/rest/dhs/indicators?returnFields=IndicatorId,La

`simplify` if TRUE only the value and region index is returned.

`admin` Either 0 (national) or 1 (subnational admin 1 area).

Value

a data frame of the DHS indicator estimates

Examples

```
## Not run:
# country:  Zambia
# survey: 2018 DHS
# indicator: Percentage of children stunted
#             (below -2 SD of height for age
#              according to the WHO standard)
dhs_table <- get_api_table(country = "ZM",
                           survey = "ZM2018DHS",
                           indicator = "CN_NUTS_C_HA2",
                           simplify = TRUE)
dhs_table

## End(Not run)
```

`hv_hiv_pos` *HA_HIVP_W_HIV hv_hiv_pos "HIV positive test result"*

Description

HA_HIVP_W_HIV hv_hiv_pos "HIV positive test result"

Usage

```
hv_hiv_pos(Rdata)
```

Arguments

`Rdata` data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "HA_HIVP_W_HIV",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::hv_hiv_pos)

## End(Not run)
```

---

intervalPlot *Get scatter plot for any two model results*

Description

This function return scatter plot at admin 1 level for any two model results

Usage

```
intervalPlot(admin = 0, compare = FALSE, model = NULL, group = FALSE)
```

Arguments

admin	level of plot
compare	plot for compare multiple plot or not
model	list of model results using surveyPrev
group	plot by group or not

Value

This function returns the dataset that contain district name and population for given tiff files and polygons of admin level.

Author(s)

Qianyu Dong

Examples

```
## Not run:

geo <- getDHSgeo(country = "Zambia", year = 2018)
data(ZambiaAdm1)
data(ZambiaAdm2)
data(ZambiaPopWomen)
cluster.info <- clusterInfo(geo = geo,
                            poly.adm1 = ZambiaAdm1,
                            poly.adm2 = ZambiaAdm2)

dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "ancvisit4+",
                                 year = 2018)

data <- getDHSindicator(dhsData, indicator = "ancvisit4+")
admin.info2 <- adminInfo(poly.adm = ZambiaAdm2,
                        admin = 2,
                        agg.pop =ZambiaPopWomen$admin2_pop,
                        proportion = ZambiaPopWomen$admin2_urban)
cl_res_ad2_unstrat <- clusterModel(data = data,
                  cluster.info = cluster.info,
                  admin.info = admin.info2,
                  stratification = FALSE,
                  model = "bym2",
                  admin = 2,
                  aggregation = TRUE,
                  CI = 0.95)

head(cl_res_ad2_unstrat$res.admin2)
head(cl_res_ad2_unstrat$agg.admin1)
plots <- intervalPlot(cl_res_ad2_unstrat)
plots[["Central"]]

cl_res_ad2 <- clusterModel(data = data,
                  cluster.info = cluster.info,
                  admin.info = admin.info2,
                  stratification = TRUE,
                  model = "bym2",
                  admin = 2,
                  aggregation = TRUE,
                  CI = 0.95)
head(cl_res_ad2$res.admin2)
head(cl_res_ad2$agg.admin1)
plots <- intervalPlot(cl_res_ad2)
plots[["Central"]]

library(patchwork)
wrap_plots(plots, ncol = 5)


## End(Not run)
```

ml_hhaccess *ML_ITNA_P_ACC Households with at least one insecticide-treated mosquito net (ITN) for every two persons who stayed in the household the previous night Persons with access to an insecticide-treated mosquito net (ITN) ML_NETS_HH.do HR Households with >1 ITN per 2 household members Percentage of households with at least one ITN for every 2 persons who stayed in the household last night*

Description

ML_ITNA_P_ACC Households with at least one insecticide-treated mosquito net (ITN) for every two persons who stayed in the household the previous night Persons with access to an insecticide-treated mosquito net (ITN) ML_NETS_HH.do HR Households with >1 ITN per 2 household members Percentage of households with at least one ITN for every 2 persons who stayed in the household last night

Usage

```
ml_hhaccess(Rdata)
```

Arguments

Rdata	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "ML_ITNA_P_ACC",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::ml_hhaccess)

## End(Not run)
```

NMR *CM_ECMR_C_NNR nmr CM_ECMR_C_NNR BR (not from dhs github) Neonatal mortality rate !!!!!!*

Description

CM_ECMR_C_NNR nmr CM_ECMR_C_NNR BR (not from dhs github) Neonatal mortality rate !!!!!!

Usage

```
NMR(Rdata, nmr.year)
```

Arguments

`Rdata`	data.frame from survryPrev::getDHSdata
`nmr.year`	This is an argument specifically for NMR calculation. It specifies births how many years do we include prior to the date of survey. Default to be 10, i.e., NMR in the last 10 years prior to survey.

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "CM_ECMR_C_NNR",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::NMR)

## End(Not run)
```

nt_ch_any_anem *CN_ANMC_C_ANY Children with any anemia "Any anemia - child 6-59 months" PR NT_CH_NUT.do Children under five with any anemia*

Description

CN_ANMC_C_ANY Children with any anemia "Any anemia - child 6-59 months" PR NT_CH_NUT.do Children under five with any anemia

Usage

```
nt_ch_any_anem(Rdata)
```

Arguments

Rdata	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                  indicator = "CN_ANMC_C_ANY",
                                  year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                          FUN = surveyPrev::nt_ch_any_anem)

## End(Not run)
```

nt_ch_stunt *CN_NUTS_C_HA2 stunting Children stunted NT_CH_NUT.do PR "Stunted child under 5 years" Stunting rate (Prevalence of stunted (HAZ < -2) children under five (0-59 months)) Percentage of children under age five stunted (below -2 standard deviations of height-for-age according to the WHO standard).*

Description

CN_NUTS_C_HA2 stunting Children stunted NT_CH_NUT.do PR "Stunted child under 5 years" Stunting rate (Prevalence of stunted (HAZ < -2) children under five (0-59 months)) Percentage of children under age five stunted (below -2 standard deviations of height-for-age according to the WHO standard).

Usage

```
nt_ch_stunt(Rdata)
```

Arguments

Rdata	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "CN_NUTS_C_HA2",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::nt_ch_stunt)

## End(Not run)
```

---

nt_ch_wast *CN_NUTS_C_WH2 wasting Children wasted NT_CH_NUT.do PR "Wasted child under 5 years" Wasting rate (Prevalence of wasted (HAZ < -2) children under five (0-59 months)) Percentage of children under age five with a weight-for-height z-score (WHZ) more than two standard deviations below the median WHO growth standards.*

Description

CN_NUTS_C_WH2 wasting Children wasted NT_CH_NUT.do PR "Wasted child under 5 years" Wasting rate (Prevalence of wasted (HAZ < -2) children under five (0-59 months)) Percentage of children under age five with a weight-for-height z-score (WHZ) more than two standard deviations below the median WHO growth standards.

Usage

```
nt_ch_wast(Rdata)
```

Arguments

`Rdata`	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "CN_NUTS_C_WH2",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::nt_ch_wast)

## End(Not run)
```

---

nt_ebf *CN_BRFS_C_EXB Children exclusively breastfed NT_IYCF.do KR "Exclusively breastfed - last-born under 6 months" Children exclusively breastfed (Prevalence of exclusive breastfeeding of children under six months of age)*

Description

CN_BRFS_C_EXB Children exclusively breastfed NT_IYCF.do KR "Exclusively breastfed - last-born under 6 months" Children exclusively breastfed (Prevalence of exclusive breastfeeding of children under six months of age)

Usage

```
nt_ebf(Rdata)
```

Arguments

`Rdata`	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                  indicator = "CN_BRFS_C_EXB",
                                  year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::nt_ebf)

## End(Not run)
```

`nt_wm_any_anem` *AN_ANEM_W_ANY womananemia nt_wm_any_anem "Any anemia - women" NT_WM_NUT.do Percentage of women aged 15-49 classified as having any anemia*

Description

AN_ANEM_W_ANY womananemia nt_wm_any_anem "Any anemia - women" NT_WM_NUT.do Percentage of women aged 15-49 classified as having any anemia

Usage

```
nt_wm_any_anem(Rdata)
```

Arguments

`Rdata`	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "AN_ANEM_W_ANY",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::nt_wm_any_anem)

## End(Not run)
```

`nt_wm_thin` *AN_NUTS_W_THN Women who are thin according to BMI (<18.5) NT_WM_NUT.do "Thin BMI - women" IR !!!!!!!! Underweight (Prevalence of underweight (BMI < 18.5) women of reproductive age)*

Description

AN_NUTS_W_THN Women who are thin according to BMI (<18.5) NT_WM_NUT.do "Thin BMI - women" IR !!!!!!!! Underweight (Prevalence of underweight (BMI < 18.5) women of reproductive age)

Usage

```
nt_wm_thin(Rdata)
```

Arguments

`Rdata`	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "AN_NUTS_W_THN",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::nt_wm_thin)

## End(Not run)
```

ph_sani_basic *WS_TLET_P_BAS Population with access to a basic sanitation service WS_TLET_P_BAS in DHS API PH_SANI.do PR ph_sani_basic "Basic sanitation facility"*

Description

WS_TLET_P_BAS Population with access to a basic sanitation service WS_TLET_P_BAS in DHS API PH_SANI.do PR ph_sani_basic "Basic sanitation facility"

Usage

```
ph_sani_basic(Rdata)
```

Arguments

`Rdata`	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "WS_TLET_P_BAS",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::ph_sani_basic)

## End(Not run)
```

ph_sani_improve *WS_TLET_H_IMP Percentage of households using an improved sanitation facility PH_SANI.do PR ph_sani_improve "Access to improved sanitation" country-specific*

Description

WS_TLET_H_IMP Percentage of households using an improved sanitation facility PH_SANI.do PR ph_sani_improve "Access to improved sanitation" country-specific

Usage

```
ph_sani_improve(Rdata)
```

Arguments

Rdata	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "WS_TLET_H_IMP",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::ph_sani_improve)

## End(Not run)
```

rankPlot *Get ranking plot of model results*

Description

This function return scatter plot at admin 1 or 2 level for given model results.

Usage

```
rankPlot(x, direction = 1)
```

Arguments

x	a model result using surveyPrev of class "fhModel" or "clusterModel"
direction	direction of the ranking. The default is direction = 1, which correspond to larger value having higher ranking. If direction = -1, larger value has lower ranking.

Value

This function returns a ranking plot.

Author(s)

Zehang Li, Qianyu Dong

Examples

```
## Not run:

geo <- getDHSgeo(country = "Zambia", year = 2018)
data(ZambiaAdm1)
data(ZambiaAdm2)
data(ZambiaPopWomen)
cluster.info <- clusterInfo(geo = geo,
                            poly.adm1 = ZambiaAdm1,
                            poly.adm2 = ZambiaAdm2)

dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "ancvisit4+",
                                 year = 2018)

data <- getDHSindicator(dhsData, indicator = "ancvisit4+")
admin.info2 <- adminInfo(poly.adm = ZambiaAdm2,
                        admin = 2,
         by.adm="NAME_2",
         by.adm.upper = "NAME_1")
cl_res_ad2_unstrat <- clusterModel(data = data,
                  cluster.info = cluster.info,
                  admin.info = admin.info2,
                  stratification = FALSE,
                  model = "bym2",
                  admin = 2,
                  aggregation = TRUE,
                  CI = 0.95)
rankPlot(cl_res_ad2_unstrat)


## End(Not run)
```

rh_anc_4vs — *RH_ANCN_W_N4P ancvisit4+ RH_ANCN_W_N4P IR Antenatal visits for pregnancy: 4+ visits*

Description

RH_ANCN_W_N4P ancvisit4+ RH_ANCN_W_N4P IR Antenatal visits for pregnancy: 4+ visits

Usage

```
rh_anc_4vs(Rdata)
```

Arguments

`Rdata`	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                  indicator = "RH_ANCN_W_N4P",
                                  year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::rh_anc_4vs)

## End(Not run)
```

rh_del_pvskill *RH_DELA_C_SKP IR or BR Assistance during delivery from a skilled provider*

Description

RH_DELA_C_SKP IR or BR Assistance during delivery from a skilled provider

Usage

```
rh_del_pvskill(Rdata)
```

Arguments

Rdata	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run:
dhsData <- getDHSdata(country = "Zambia",
                                  indicator = "RH_DELA_C_SKP",
                                  year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::rh_del_pvskill)

## End(Not run)
```

scatterPlot *Get scatter plot for any two model results*

Description

This function return scatter plot at admin 1 level for any two model results

Usage

```
scatterPlot(
  res1,
  value1,
  res2,
  value2,
  label1,
  label2,
  by.res1,
  by.res2,
  title
)
```

Arguments

`res1`	model result 1 using surveyPrev
`value1`	value1
`res2`	model result 2 using surveyPrev
`value2`	value2
`label1`	label for x axis
`label2`	label for y axis
`by.res1`	by.res1
`by.res2`	by.res2
`title`	title

Value

This function returns the dataset that contain district name and population for given tiff files and polygons of admin level

Author(s)

Qianyu Dong

Examples

```
## Not run:
geo <- getDHSgeo(country = "Zambia", year = 2018)
data(ZambiaAdm1)
data(ZambiaAdm2)
data(ZambiaPopWomen)
cluster.info <- clusterInfo(geo = geo,
                            poly.adm1 = ZambiaAdm1,
                            poly.adm2 = ZambiaAdm2)

dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "ancvisit4+",
                                 year = 2018)

data <- getDHSindicator(dhsData, indicator = "ancvisit4+")
admin.info1 <- adminInfo(poly.adm = ZambiaAdm1,
                        admin = 1,
                        agg.pop =ZambiaPopWomen$admin1_pop,
                        proportion = ZambiaPopWomen$admin1_urban)
smth_res_ad1 <- fhModel(data,
                       cluster.info = cluster.info,
                       admin.info = admin.info1,
                       admin = 1,
                       model = "bym2",
                       aggregation = F)

admin.info2 <- adminInfo(poly.adm = ZambiaAdm2,
                        admin = 2,
                        agg.pop =ZambiaPopWomen$admin2_pop,
                        proportion = ZambiaPopWomen$admin2_urban)
cl_res_ad2 <- clusterModel(data = data,
                  cluster.info = cluster.info,
                  admin.info = admin.info2,
                  stratification = FALSE,
                  model = "bym2",
                  admin = 2,
                  aggregation = TRUE,
                  CI = 0.95)

scatterPlot(
     res1 = smth_res_ad1,
     res2 = cl_res_ad2$agg.admin1,
     value1 = "value",
     value2 = "value",
     by.res1 = "admin1.name",
     by.res2 = "admin1.name",
     title = "Aggregated cluster model v.s. Fay-Herriot",
     label1 = "Fay-Herriot",
     label2 = "Aggregated cluster model")


## End(Not run)
```

surveyPrevIndicators *Table of built-in indicators.*

Description

A data frame of indicators currently implemented in the package

Usage

```
data(surveyPrevIndicators)
```

Format

An object of class `data.frame` with 22 rows and 4 columns.

---

watersource_adj *WS_SRCE_P_BAS Population using a basic water source PH_WATER.do ph_wtr_basic "Basic water service" PR*

Description

WS_SRCE_P_BAS Population using a basic water source PH_WATER.do ph_wtr_basic "Basic water service" PR

Usage

```
watersource_adj(Rdata)
```

Arguments

`Rdata`	data.frame from survryPrev::getDHSdata

Value

A partially processed data.frame that will be used in survryPrev::getDHSindicator. The whole function can be used as a parameter in survryPrev::getDHSindicator

Author(s)

Qianyu Dong

Examples

```
## Not run: 
dhsData <- getDHSdata(country = "Zambia",
                                 indicator = "WS_SRCE_P_BAS",
                                 year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                         FUN = surveyPrev::watersource_adj)

## End(Not run)
```

ZambiaAdm1 *Admin 1 Polygon Map for Zambia.*

Description

A SpatialPolygonsDataFrame corresponding to Zambia's admin-1 regions. The dataset is downloaded from GADM (https://gadm.org/data.html) version 4.1.

Usage

```
data(ZambiaAdm1)
```

Format

An object of class `SpatialPolygonsDataFrame` with 10 rows and 11 columns.

ZambiaAdm2 *Admin 2 Polygon Map for Zambia.*

Description

A SpatialPolygonsDataFrame corresponding to Zambia's admin-2 regions. The dataset is downloaded from GADM (https://gadm.org/data.html) version 4.1.

Usage

```
data(ZambiaAdm2)
```

Format

An object of class `SpatialPolygonsDataFrame` with 115 rows and 13 columns.

ZambiaPopWomen *Population estimates for Women of age 15 to 49 in Zambia in 2018.*

Description

A list of three objects

- raster A 100m by 100m raster file for the population estimates for women of age 15 to 49 in Zambia in 2018
- admin1_urban A data frame specifying the proportion of urban population (as defined by those living in regions designated as urban in the previous census) for Women of age 15 to 49 in Zambia in 2018, in each admin1 region.
- admin2_urban A data frame specifying the proportion of urban population (as defined by those living in regions designated as urban in the previous census) for Women of age 15 to 49 in Zambia in 2018, in each admin2 region. The corresponding admin1 region name is also included.

The dataset is downloaded from WorldPop (https://hub.worldpop.org/geodata/summary?id=16429) and post processed.

Usage

```
data(ZambiaPopWomen)
```

Format

An object of class `list` of length 4.

Index