

# Package ‘surveyPrev’

April 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](https://doi.org/10.48550/arXiv.2110.09576)> and Wakefield et al. (2020) <[doi:10.1111/insr.12400](https://doi.org/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, rdhs, SUMMER, dplyr, labelled, sjlabelled, naniar, raster, sp, spdep, stringr, tidyverse, data.table, sf, matrixStats

**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

**Author** Qianyu Dong [cre, aut],  
Zehang R Li [aut],  
Yunhan Wu [aut],  
Andrea Boskovic [aut],  
Jon Wakefield [aut]

**Maintainer** Qianyu Dong <qdong14@ucsc.edu>

Repository CRAN

Date/Publication 2024-04-10 19:50:02 UTC

## R topics documented:

adminInfo . . . . .	3
aggPopulation . . . . .	4
aggSurveyWeight . . . . .	5
ch_allvac_either . . . . .	7
ch_diar_ors_rhf . . . . .	8
ch_meas_either . . . . .	8
ch_novac_either . . . . .	9
ch_pent1_either . . . . .	10
ch_pent3_either . . . . .	11
clusterInfo . . . . .	11
clusterModel . . . . .	12
directEST . . . . .	14
fhModel . . . . .	16
fp_cruse_mod . . . . .	17
fp_unmet_tot . . . . .	18
getDHSdata . . . . .	19
getDHSgeo . . . . .	20
getDHSindicator . . . . .	21
getUR . . . . .	22
get_api_table . . . . .	24
hv_hiv_pos . . . . .	25
intervalPlot . . . . .	26
ml_hhaccess . . . . .	27
NMR . . . . .	28
nt_ch_any_anem . . . . .	29
nt_ch_stunt . . . . .	30
nt_ch_wast . . . . .	31
nt_ebf . . . . .	32
nt_wm_any_anem . . . . .	33
nt_wm_thin . . . . .	34
ph_sani_basic . . . . .	35
ph_sani_improve . . . . .	36
rh_anc_4vs . . . . .	37
rh_del_pvskill . . . . .	37
scatterPlot . . . . .	38
surveyPrevIndicators . . . . .	40
watersource_adj . . . . .	41
ZambiaAdm1 . . . . .	41
ZambiaAdm2 . . . . .	42
ZambiaPopWomen . . . . .	42

Index

44

---

adminInfo	<i>Get admin information</i>
-----------	------------------------------

---

### Description

This function get admin information including name, character, population and urban/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 <code>sp::SpatialPolygonsDataFrame</code> object or an <code>sf</code> 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 population from <code>aggPopulation</code> function. It should have two columns: "admin2.name.full" and "population".
proportion	data frame of urban/rural proportions. For admin1, it 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".
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	<i>CH_VACS_C_BAS Children with all 8 basic vaccinations (age 12-23) "All basic vaccinations according to either source"</i>
------------------	---

---

### 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 = survryPrev::ch_allvac_either)

## End(Not run)

```

---

ch_diar_ors_rhf	<i>CH_DIAT_C_ORT KR Diarrhea treatment (Children under five with diarrhea treated with either ORS or RHF)</i>
-----------------	---

---

**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	<i>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"</i>
----------------	--

---

**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	<i>CH_VACS_C_NON KR Children with no vaccinations (age 12-23)</i>
-----------------	---

---

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

---

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

---

**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	<i>Get cluster information</i>
-------------	--------------------------------

---

**Description**

This function add admin 1 and admin2 information to a particular 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,
  admin,
  CI = 0.95,
  model = c("bym2", "iid"),
  stratification = FALSE,
  aggregation = FALSE,
```

```

    overdisp.mean = 0,
    overdisp.prec = 0.4
  )

```

### Arguments

data	dataframe that contains the indicator of interests, 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
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.

### 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,
  ...
)

```

## 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.
...	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=poly.adm1, poly.adm2=poly.adm2,
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")
```

```
## 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,
  admin,
  CI = 0.95,
  model = c("bym2", "iid"),
  aggregation = FALSE
)
```

## Arguments

<code>data</code>	dataframe that contains the indicator of interests, output of <code>getDHSindicator</code> function
<code>cluster.info</code>	list contains data and <code>wrong.points</code> . data contains admin 1 and admin 2 information and coordinates for each cluster. <code>wrong.points</code> . contains cluster id for cluster without coordinates or admin 1 information. Output of <code>getDHSindicator</code> function
<code>admin.info</code>	list contains data and <code>mat</code> , data contains population and urban/rural proportion at specific admin level and <code>mat</code> is the adjacency matrix, output of <code>adminInfo</code> function
<code>admin</code>	admin level for the model
<code>CI</code>	Credible interval to be used. Default to 0.95.
<code>model</code>	smoothing model used in the random effect. Options are independent ("iid") or spatial ("bym2").
<code>aggregation</code>	whether or not report aggregation results.

## 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	<i>FP_CUSM_W_MOD IRdata Modern contraceptive prevalence rate (Married women currently using any modern method of contraception)</i>
--------------	---

---

**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_CUSM_W_MOD",
                      year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                       FUN = survryPrev::fp_cruse_mod)

## End(Not run)
```

---

fp_unmet_tot	<i>FP_NADA_W_UNT #unmet_family IRdata women with an unmet need for family planning for spacing and limiting</i>
--------------	---

---

**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,
```

```
names(dhsData)          year = 2018)
## End(Not run)
```

---

getDHSgeo	<i>Download DHS geo data</i>
-----------	------------------------------

---

### 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	<i>Process DHS data</i>
-----------------	-------------------------

---

**Description**

This function processes DHS data from getDHSdata function.

**Usage**

```
getDHSindicator(Rdata, indicator = NULL, FUN = NULL)
```

**Arguments**

Rdata	Result from getDHSdata function, the raw DHS survey 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.

**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")
```

```
## 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_m_0_2018.tif"

# Since the two files are very large, we recommend downloading them
# manually 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)
```

## Arguments

country	A character string of keys at: <a href="https://api.dhsprogram.com/rest/dhs/countries?returnFields=CountryName">https://api.dhsprogram.com/rest/dhs/countries?returnFields=CountryName</a> .
survey	A character string of keys at: <a href="https://api.dhsprogram.com/rest/dhs/surveys?returnFields=SurveyId">https://api.dhsprogram.com/rest/dhs/surveys?returnFields=SurveyId</a> .
indicator	A character string of keys at: <a href="https://api.dhsprogram.com/rest/dhs/indicators?returnFields=IndicatorId">https://api.dhsprogram.com/rest/dhs/indicators?returnFields=IndicatorId</a> .
simplify	if TRUE only the value and region index is returned.



**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	<i>HA_HIVP_W_HIV hv_hiv_pos "HIV positive test result"</i>
------------	--

---

**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**

```
m1_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::m1_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)
```

**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 = "CM_ECMR_C_NNR",
                      year = 2018)
data <- getDHSindicator(dhsData, indicator = NULL,
                       FUN = survryPrev::NMR)

## End(Not run)
```

---

nt_ch_any_anem	<i>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</i>
----------------	--

---

**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	<i>CN_NUTS_C_HA2 stunting Children stunted NT_CH_NUT.do PR "Stunted child under 5 years" Stunting rate (Prevalence of stunted (HAZ &lt; -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).</i>
-------------	--

---

**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	<i>CN_NUTS_C_WH2 wasting Children wasted NT_CH_NUT.do PR "Wasted child under 5 years" Wasting rate (Prevalence of wasted (HAZ &lt; -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.</i>
------------	---

---

### 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	<i>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)</i>
--------	---

---

### 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 = survryPrev::nt_ebf)

## End(Not run)
```



---

nt_wm_any_anem	<i>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</i>
----------------	--

---

### 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 = survryPrev::nt_wm_any_anem)  
  
## End(Not run)
```

---

nt_wm_thin	<i>AN_NUTS_W_THN Women who are thin according to BMI (&lt;18.5) NT_WM_NUT.do "Thin BMI - women" IR !!!!!!! Underweight (Prevalence of underweight (BMI &lt; 18.5) women of reproductive age)</i>
------------	--

---

### 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 = survryPrev::nt_wm_thin)

## End(Not run)
```

---

ph_sani_basic	<i>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"</i>
---------------	--

---

### 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 = survryPrev::ph_sani_basic)

## End(Not run)
```

---

ph_sani_improve	<i>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</i>
-----------------	--

---

**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 = survryPrev::ph_sani_improve)

## End(Not run)
```

---

rh_anc_4vs	<i>RH_ANCN_W_N4P ancvisit4+ RH_ANCN_W_N4P IR Antenatal visits for pregnancy: 4+ visits</i>
------------	--

---

**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 = survryPrev::rh_anc_4vs)

## End(Not run)
```

---

rh_del_pvskill	<i>RH_DELA_C_SKP IR or BR Assistance during delivery from a skilled provider</i>
----------------	--

---

**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 = survryPrev::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. FayHerriot",
  label1 = "FayHerriot",
  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	<i>WS_SRCE_P_BAS Population using a basic water source PH_WATER.do ph_wtr_basic "Basic water service" PR</i>
-----------------	--

---

**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	<i>Admin 2 Polygon Map for Zambia.</i>
------------	--

---

**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	<i>Population estimates for Women of age 15 to 49 in Zambia in 2018.</i>
----------------	--

---

**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

## \* datasets

- surveyPrevIndicators, [40](#)
- ZambiaAdm1, [41](#)
- ZambiaAdm2, [42](#)
- ZambiaPopWomen, [42](#)

- adminInfo, [3](#)
- aggPopulation, [4](#)
- aggSurveyWeight, [5](#)

- ch\_allvac\_either, [7](#)
- ch\_diar\_ors\_rhf, [8](#)
- ch\_meas\_either, [8](#)
- ch\_novac\_either, [9](#)
- ch\_pent1\_either, [10](#)
- ch\_pent3\_either, [11](#)
- clusterInfo, [11](#)
- clusterModel, [12](#)

- directEST, [14](#)

- fhModel, [16](#)
- fp\_cruse\_mod, [17](#)
- fp\_unmet\_tot, [18](#)

- get\_api\_table, [24](#)
- getDHSdata, [19](#)
- getDHSgeo, [20](#)
- getDHSindicator, [21](#)
- getUR, [22](#)

- hv\_hiv\_pos, [25](#)

- intervalPlot, [26](#)

- ml\_hhaccess, [27](#)

- NMR, [28](#)
- nt\_ch\_any\_anem, [29](#)
- nt\_ch\_stunt, [30](#)
- nt\_ch\_wast, [31](#)

- nt\_ebf, [32](#)

- nt\_wm\_any\_anem, [33](#)
- nt\_wm\_thin, [34](#)

- ph\_sani\_basic, [35](#)
- ph\_sani\_improve, [36](#)

- rh\_anc\_4vs, [37](#)
- rh\_del\_pvskill, [37](#)

- scatterPlot, [38](#)
- surveyPrevIndicators, [40](#)

- watersource\_adj, [41](#)

- ZambiaAdm1, [41](#)
- ZambiaAdm2, [42](#)
- ZambiaPopWomen, [42](#)