Package 'SocEpi'

May 26, 2020

Type Package

Title Health Inequality Analysis

Version 1.0.0

Description Tools for calculating deprivation measures and analysing health outcomes and mortality by deprivation.
Included are the functions: zscore() and w_pcntile() for developing deprivation measures and the functions smr(),
st_rate() and rii() for calculating standardized mortality ratios, direct standardized rates and Slope and Relative
indices of Inequality (SII and RII). Test data is included (dep_data and health_data).
The RII/SII are calculated following Pamuk ER (1985) <doi:10.1080/00324720310001412 56>.
The confidence intervals for SII/RII are calculated using a multinomial distribution as described in Lumme et al (2015)
<doi:10.1007/s10742-015-0137-1>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.0

Depends R (>= 3.2.0)

Imports tidyr, dplyr, Rcpp

LinkingTo Rcpp

Suggests knitr, rmarkdown, testthat

VignetteBuilder knitr

NeedsCompilation yes

Author Mirjam Allik [aut, cre]

Maintainer Mirjam Allik <mirjam.allik@gmail.com>

Repository CRAN

Date/Publication 2020-05-26 14:20:06 UTC

R topics documented:

`dep_data` *Scottish deprivation data*

Description

The data set provides the Carstairs deprivation scores, deciles, quintiles and individual Carstairs score variables for Scottish postcode sectors in 2011. A few self-rated health variables from the 2011 Scottish Census are also provided.

Usage

```
dep_data
```

Format

A data frame with 1012 rows and 26 variables:

PS_code postcode sector code

PS_name postcode sector name

health_board health board name

council_area Council area name

ur2fold Two-fold urban-rural classification, derived based on 2011-2012 Scottish Government urban-rural classification for output areas. Used in this data set for illustration only.

total_pop Number of people in postcode sector

pcnt_overcrowding Percent of people living in overcrowded households

pcnt_male_unemp Percent of unemployed men

pcnt_no_car Percent of people in households with no access to a car/van

pcnt_low_class Percent of people in low social class

carstairs The 2011 Carstairs score

pcnt_limited_lot Percent of people whose day-to-day activities are limited a lot

pcnt_no_llti Percent of people whose day-to-day activities are not limited

pcnt_bad_health Percent of people in bad or very bad health

pcnt_good_health Percent of people in good or very good health

quintile Quintiles of the Carstairs score; 1 - least deprived, 5 - most deprived

decile Deciles of the Carstairs score; 1 - least deprived, 10 - most deprived

good_health Number of people in very good and good health

fair_health Number of people in fair health

bad_health Number of people in bad or very bad health

Source

For Carstairs deprivation data look at http://www.sphsu.mrc.ac.uk/carstairs_2011_report_final_v3_-update-06-16-.pdf

Self-rated health data are from the 2011 census http://www.scotlandscensus.gov.uk/

`health_data` *Scottish health and deprivation data*

Description

A data set on deprivation and self-rated health outcomes for postcode sectors from 2011 Scottish Census by ethnicity.

Usage

```
health_data
```

Format

A data frame with 182160 rows and 15 variables:

PS_code postcode sector code

PS_name postcode sector name

health_board health board name

council_area Council area name

ur2fold Two-fold urban rural classification (created based on output area level data). Used in this data set for illustration only.

total_pop Total population of the postcode sector

carstairs The 2011 Carstairs score

quintile Quintiles of the Carstairs score; 1 - least deprived, 5 - most deprived

decile Deciles of the Carstairs score; 1 - least deprived, 10 - most deprived

ethnicity Ethnic group, "all" for all people

age age group, 1 - "0-4", 2 - "5-9", up to 18 - "85 and above"

bad number of people in bad general health

fg number of people in fairly good health

good number of people in good health

pop number of people in age and ethnic group in postcode sector

Source

For Carstairs deprivation data look at http://www.sphsu.mrc.ac.uk/carstairs_2011_report_final_v3_-update-06-16-.pdf

For health data see the 2011 census commissioned tables http://www.scotlandscensus.gov.uk/ods-web/data-warehouse.html#additionaltab

---

`rii` *RII/SII and confidence intervals*

Description

`rii` calculates the relative index of inequality (RII) or the slope index of inequality (SII) and confidence intervals for either measure. The SII is obtained via OLS regression of the health variable on the midpoints of the cumulative population distribution. The RII is calculated as SII divided by health outcome across all socioeconomic positions. Simulation is used to calculate confidence intervals (see `method` below).

Usage

```
rii(
  data,
  health,
  population,
  ses,
  age,
  groups = NULL,
  age_group = NULL,
  st_pop = "esp2013_18ag",
  N = 1000,
  RII = TRUE,
  CI = 95,
  method = "multinomial",
  W = FALSE,
  total = 1000
)
```

Arguments

`data`	Name of data set.
`health`	Health outcome of interest.
`population`	Population counts. Should correspond to data provided in `health`.
`ses`	Categorical deprivation measure used for splitting the data. Should be ordered, such as numeric or a factor.

age	Variable that defines 5-year age groups, should be ordered and numeric, such as 1 through 18. If the age group coding starts with 0 it will be assumed that age groups 0 and 1-4 are separate. If the age group starts with 1, it will be assumed that the first age group is 0-4 (ages 0 and 1-4 are combined).
groups	Conditions such as sex or ethnicity that together define the sub-population for which the RII/SII is calculated for. Default is NULL, no sub-population is selected.
age_group	The age groups the standardized rates should be calculated for. By default the function calculates results for the following age groups: 0-14, 15-29, 30-44, 45-59, 60-74, 75+, 0-64 and all ages. User supplied age groups should be provided using the standard population groups as cut-offs, e.g. use age_group=c("20-29","30-39") and not c("19-30","31-41"). Open ended age groups can be supplied by giving a single age, e.g. "45" means 45 and above. Overlapping age groups, such as c("20-29","25-34"), are not supported. Results for ages 0-64 and all ages will always be provided.
st_pop	The standard population weights used for calculating rates, default 2013 ESP for 18 age groups with 0-4 as the first age group. See st_pop for other predefined options. Can be user supplied, but must match the number of age groups given in age and add up to 1.
N	Number of simulations for the confidence intervals, default 1000.
RII	Logical, should RII or SII be calculated, default RII=TRUE.
CI	Confidence intervals, 95 by default but can be set to any number between 0 and 100.
method	The method used for simulating confidence intervals. Default method = "multinomial". The CI are calculated using a multinomial distribution as described in Lumme et al (2015) "A Monte Carlo method to estimate the confidence intervals for the concentration index using aggregated population register data." Health Services and Outcomes Research Methodology, 15(2), 82-98. http://doi.org/10.1007/s10742-015-0137-1
W	Logical, should weighted regression be used for RII/SII, default W=FALSE.
total	The total number of people in the standard population, i.e. are rates be calculated per 1000 or 100 000, default 1000. Relevant for SII only.

Value

A data frame giving RII/SII by age groups together with confidence intervals

Examples

```
d <- health_data

# RII with 95% CI
rii(d, bad, pop, quintile, age, ethnicity == "all")

# SII with 99% CI, using weighted least squares
rii(d, bad, pop, quintile, age, ethnicity == "all", RII = FALSE, CI = 99, W = TRUE)
```

```
# Supply own population weights
new_w <- c(0.075, 0.075, 0.075, 0.06, 0.060, 0.060, 0.06, 0.070, 0.050,
   0.050, 0.050, 0.06, 0.060, 0.055, 0.050, 0.040, 0.025, 0.025)

# RII with user supplied weights
rii(d, bad, pop, quintile, age, ethnicity == "all", RII = FALSE, CI = 99, st_pop = new_w)

# SII for new age groups with 95% CI
rii(d, bad, pop, quintile, age, ethnicity == "Scot" &  ur2fold == "Urban",
   age_group=c("0-19", "20-34", "35-49"), RII = FALSE)
```

`smr` *Standardized mortality ratios and confidence intervals*

Description

The function `smr` calculates standardized mortality ratios (SMR) and confidence intervals for SMR.

Usage

```
smr(data, health, population, age, compare, sets, age_group = NULL, CI = 95)
```

Arguments

`data`	Name of data set.
`health`	Health outcome of interest.
`population`	Population counts. Should correspond to data provided in `health`.
`age`	Variable that defines 5-year age groups, should be ordered and numeric, such as 1 through 18. If the age group coding starts with 0 it will be assumed that age groups 0 and 1-4 are separate. If the age group starts with 1, it will be assumed that the first age group is 0-4 (ages 0 and 1-4 are combined).
`compare`	Categorical variable that splits the data into groups that are to be compared, such as ethnicity.
`sets`	Groups that are to be compared (values taken by `compare`). The group listed first will be the reference category. Must take at least two values.
`age_group`	The age groups the standardized rates should be calculated for. By default the function calculates results for the following age groups: 0-14, 15-29, 30-44, 45-59, 60-74, 75+, 0-64 and all ages. User supplied age groups should be provided using the standard population groups as cut-offs, e.g. use `age_group=c("20-29","30-39")` and not `c("19-30","31-41")`. Open ended age groups can be supplied by giving a single age, e.g. "45" means 45 and above. Overlapping age groups, such as `c("20-29","25-34")`, are not supported. Results for ages 0-64 and all ages will always be provided.
`CI`	Confidence intervals, 95 by default but can be set to any number between 0 and 100. For calculation method see user guide.

Value

A data frame of standardized mortality ratios (SMR) and CIs.

Examples

```
d <- health_data

# Asian population compared to Scottish (reference)
smr(d, bad, pop, age, ethnicity, sets = c("Scot", "asian"))

# Asian, White British and Irish population compared to Scottish (reference)
smr(d, bad, pop, age, ethnicity, sets = c("Scot", "asian", "WB", "Irish"),
  age_group = c("15-29", "30-44"), CI = 99)
```

---

st_pop *Standard population data*

Description

Functions rii and st_rate use standard population weights to calculate direct standardized rates for health outcomes. A number of more frequently used standard population weights come with the package. The below table describes these weights.

Details

age	esp2013_18ag	esp2013_19ag	esp2013_20ag	esp2013_21ag	esp1976_18ag	esp1976_19ag	who2025_18
0				0.010		0.016	
1-4	0.050	0.050	0.050	0.040	0.08	0.064	0.08
5-9	0.055	0.055	0.055	0.055	0.07	0.070	0.08
10-14	0.055	0.055	0.055	0.055	0.07	0.070	0.08
15-19	0.055	0.055	0.055	0.055	0.07	0.070	0.08
20-24	0.060	0.060	0.060	0.060	0.07	0.070	0.08
25-29	0.060	0.060	0.060	0.060	0.07	0.070	0.07
30-34	0.065	0.065	0.065	0.065	0.07	0.070	0.07
35-39	0.070	0.070	0.070	0.070	0.07	0.070	0.07
40-44	0.070	0.070	0.070	0.070	0.07	0.070	0.06
45-49	0.070	0.070	0.070	0.070	0.07	0.070	0.06
50-54	0.070	0.070	0.070	0.070	0.07	0.070	0.05
55-59	0.065	0.065	0.065	0.065	0.06	0.060	0.04
60-64	0.060	0.060	0.060	0.060	0.05	0.050	0.03
65-69	0.055	0.055	0.055	0.055	0.04	0.040	0.02
70-74	0.050	0.050	0.050	0.050	0.03	0.030	0.02
75-79	0.040	0.040	0.040	0.040	0.02	0.020	0.01
80-84	0.025	0.025	0.025	0.025	0.01	0.010	0.00
85-89	0.025	0.015	0.015	0.015	0.01	0.010	0.00
90-94		0.010	0.008	0.008			
95-99			0.002	0.002			
100 +							

st_rate *Standardized mortality rates and confidence intervals*

Description

st_rate calculates direct standardized mortality rates and the 95% or 99% confidence intervals. Default standardization uses the 2013 European Standard Population, but other built-in or user supplied weights can also be supplied.

Usage

```
st_rate(
  data,
  health,
  population,
  ses,
  age,
  groups = NULL,
  age_group = NULL,
  st_pop = "esp2013_18ag",
  CI = 95,
  total = 1000
)
```

Arguments

data	Name of data set.
health	Health outcome of interest.
population	Population counts. Should correspond to data provided in health.
ses	Categorical deprivation measure used for splitting the data.
age	Variable that defines 5-year age groups, should be ordered and numeric, such as 1 through 18. If the age group coding starts with 0 it will be assumed that age groups 0 and 1-4 are separate. If the age group starts with 1, it will be assumed that the first age group is 0-4 (ages 0 and 1-4 are combined).
groups	Conditions such as sex or ethnicity that together define the sub-population for which the rates are calculated for. Default is NULL, no sub-population is selected.
age_group	The age groups the standardized rates should be calculated for. By default the function calculates results for the following age groups: 0-14, 15-29, 30-44, 45-59, 60-74, 75+, 0-64 and all ages. User supplied age groups should be provided using the standard population groups as cut-offs, e.g. use age_group=c("20-29","30-39") and not c("19-30","31-41"). Open ended age groups can be supplied by giving a single age, e.g. "45" means 45 and above. Overlapping age groups, such as c("20-29","25-34"), are not supported. Results for ages 0-64 and all ages will always be provided.

st_pop	The standard population weights used for calculating rates, default 2013 ESP for 18 age groups with 0-4 as the first age group. See st_pop for other predefined options. Can be user supplied, but must match the number of age groups given in age and add up to 1.
CI	Confidence intervals, 95 by default but can be set to any number between 0 and 100.
total	The total number of people in the standard population, i.e. are rates be calculated per 1000 or 100 000, default 1000

Value

A data frame giving standardized rates by age group and deprivation together with CI.

Examples

```
d <- health_data

# Standardized rates for all people
st_rate(d, bad, pop, quintile, age, ethnicity == "all")

# Or save results
rate_data <- st_rate(d, bad, pop, quintile, age, ethnicity == "all")
# Then use View(rate_data) to view results

# Standardized rates for Scottish, with 99% CI
st_rate(d, bad, pop, quintile, age, ethnicity == "Scot", age_group = c("15-29", "30-44"), CI = 99)
```

---

w_pcntile *Function to create population weighted deprivation percentiles*

Description

Function to create population weighted deprivation percentiles

Usage

```
w_pcntile(data, population, variable, p = 10, low = FALSE)
```

Arguments

data	name of the dataset
population	distribution (in numbers)
variable	the deprivation measure to calculate the percentiles for
p	the number of percentiles/groups to split the data in. Default is deciles p=10.
low	should low values of the percentile correspond to high deprivation, defaults to FALSE – higher values correspond to higher deprivation

Value

Numeric vector of percentiles (default 1 to 10) of same length as `population` and `variable`

Examples

```
data <- dep_data

#calculate deciles for overcrowding
data$dec_overcrowd <- w_pcntile(data, total_pop, pcnt_overcrowding)

#average percent of overcrowding by decile
tapply(data$pcnt_overcrowding, data$dec_overcrowd, mean)

#percent of people in each decile
round(tapply(data$total_pop, data$dec_overcrowd, sum)/sum(data$total_pop)*100, 1)

#calculate quintiles from deciles
data$Q_overcrowd <- cut(data$dec_overcrowd, breaks = 5, labels = 1:5)

#calculate quintiles with w_pcntile
data$Q_overcrowd2 <- w_pcntile(data, total_pop, pcnt_overcrowding, p = 5)

#compare results - note small differences
table(data$Q_overcrowd, data$Q_overcrowd2)
```

---

zscore *z-scores*

Description

`zscores` calculates standardized scores or z-scores, with a mean close to zero and standard deviation (sd) close to one, for the variable given in `variable`. The standardization uses a population weighted mean and standard deviation, which are calculated based on the population distribution given by `population`.

Usage

```
zscore(population, variable)
```

Arguments

`population`	Population counts.
`variable`	Continuous deprivation measure, such as a percentage, to calculate the z-scores for.

Value

A list including the following:

`z.score`	standardized score using weighted mean and sd
`w.mean`	weighted mean
`w.sd`	weighted standard deviation
`weight`	population weight

Examples

```
data <- dep_data

# store all results in object z_oc
z_oc <- zscore(data$total_pop, data$pcnt_overcrowding)

# extract z-score
data$z_overcrowd <- z_oc$z.score
mean(data$z_overcrowd) # mean of z-score
sd(data$z_overcrowd) # sd z-score

# extract weighted mean and sd, compare weighted values to unweighted values
mean(data$pcnt_overcrowding)
z_oc$w.mean
sd(data$pcnt_overcrowding)
z_oc$w.sd
```

Index