Package ‘healthiar’

May 8, 2026

Title Quantifying and Monetizing Health Impacts Attributable to Exposure

Version 0.2.4

Description This R package has been developed with a focus on air pollution and noise but can be applied to other exposures. The initial development has been funded by the European Union project BEST-COST. Disclaimer: It is work in progress and the developers are not liable for any calculation errors or inaccuracies resulting from the use of this package.
Selection of relevant references (in chronological order):
WHO (2003) <`https://www.who.int/publications/i/item/9241546204`>,
Murray et al. (2003) <doi:10.1186/1478-7954-1-1>,
Miller & Hurley (2003) <doi:10.1136/jech.57.3.200>,
Steenland & Armstrong (2006) <doi:10.1097/01.ede.0000229155.05644.43>,
WHO (2011) <`https://iris.who.int/items/723ab97c-5c33-4e3b-8df1-744aa5bc1c27`>,
GBD 2019 Risk Factors Collaborators (2020) <doi:10.1016/S0140-6736(20)30752-2>.

License GPL (>= 3)

URL `https://swisstph.github.io/healthiar/`,
`https://github.com/SwissTPH/healthiar`,
`https://doi.org/10.5281/zenodo.18220360`

BugReports `https://github.com/SwissTPH/healthiar/issues`

Encoding UTF-8

RoxygenNote 7.3.3

Imports dplyr, tidyr, purrr, tibble, Rdpack

RdMacros Rdpack

Depends R (>= 4.2.0)

LazyData true

Config/testthat/edition 3

VignetteBuilder knitr

Suggests sf, terra, exactextractr, testthat, devtools, knitr, rmarkdown

Config/Needs/website rmarkdown

NeedsCompilation no

Author Alberto Castro [cre, aut] (ORCID: <https://orcid.org/0000-0002-4665-3299>),
Axel Luyten [aut] (ORCID: <https://orcid.org/0000-0002-7005-5889>),
Arno Pauwels [ctb] (ORCID: <https://orcid.org/0000-0001-7519-8080>),
Liliana Vazquez Fernandez [ctb] (ORCID: <https://orcid.org/0000-0003-3778-9415>),
Gianni Ardielli [ctb] (ORCID: <https://orcid.org/0009-0000-3408-2217>),
Iracy Pimenta [ctb] (ORCID: <https://orcid.org/0000-0003-0032-1536>),
Susanne Breitner [ctb] (ORCID: <https://orcid.org/0000-0002-0956-6911>),
Carl Baravelli [ctb] (ORCID: <https://orcid.org/0000-0001-7772-5315>),
Vanessa Gorasso [ctb] (ORCID: <https://orcid.org/0000-0001-6884-9316>),
Maria Lepnurm [ctb] (ORCID: <https://orcid.org/0009-0009-4372-6227>),
Andreia Novais [ctb] (ORCID: <https://orcid.org/0009-0007-7775-108X>),
Maria Jose Rueda Lopez [ctb] (ORCID: <https://orcid.org/0000-0002-2443-1038>),
Minh Nhat Pham [ctb] (ORCID: <https://orcid.org/0000-0002-5972-1668>),
Ana Barbosa [ctb] (ORCID: <https://orcid.org/0000-0002-9623-9002>),
Joao Vasco Santos [ctb] (ORCID: <https://orcid.org/0000-0003-4696-1002>),
Anette Kocbach Bolling [ctb] (ORCID: <https://orcid.org/0000-0003-4209-7448>)

Maintainer Alberto Castro <alberto.castrofernandez@swisstph.ch>

Repository CRAN

Date/Publication 2026-03-12 09:00:14 UTC

Contents

`attribute_health` *Attribute health impacts to an environmental stressor*

Description

This function calculates the attributable health impacts (mortality or morbidity) due to exposure to an environmental stressor (air pollution or noise), using either relative risk (**RR**) or absolute risk (**AR**).

Arguments for both **RR & AR** pathways

- `approach_risk`
- `exp_central`, `exp_lower`, `exp_upper`
- `cutoff_central`, `cutoff_lower`, `cutoff_upper`
- `erf_eq_central`, `erf_eq_lower`, `erf_eq_upper`

Arguments only for **RR** pathways

- `rr_central`, `rr_lower`, `rr_upper`
- `rr_increment`
- `erf_shape`
- `bhd_central`, `bhd_lower`, `bhd_upper`
- `prop_pop_exp`

Argument for **AR** pathways

- `pop_exp`

Optional arguments for both **RR & AR** pathways

- `geo_id_micro`, `geo_id_macro`,
- `age_group`, `sex`, `info`, `population`
- `dw_central`, `dw_lower`, `dw_upper`
- `duration_central`, `duration_lower`, `duration_upper`

Usage

```
attribute_health(
  approach_risk = "relative_risk",
  exp_central,
  exp_lower = NULL,
  exp_upper = NULL,
  cutoff_central = 0,
  cutoff_lower = NULL,
  cutoff_upper = NULL,
  pop_exp = NULL,
  erf_eq_central = NULL,
  erf_eq_lower = NULL,
  erf_eq_upper = NULL,
  rr_central = NULL,
  rr_lower = NULL,
  rr_upper = NULL,
  rr_increment = NULL,
  erf_shape = NULL,
  bhd_central = NULL,
  bhd_lower = NULL,
  bhd_upper = NULL,
  prop_pop_exp = 1,
  geo_id_micro = "a",
  geo_id_macro = NULL,
  age_group = "all",
  sex = "all",
  dw_central = NULL,
  dw_lower = NULL,
  dw_upper = NULL,
  duration_central = NULL,
  duration_lower = NULL,
  duration_upper = NULL,
  info = NULL,
  population = NULL
)
```

Arguments

`approach_risk`	`String` `value` specifying the **risk method**. Options: `"relative_risk"` (default) or `"absolute_risk"`.
`exp_central`, `exp_lower`, `exp_upper`	`Numeric` `value` or `numeric` `vector` specifying the **exposure level(s)** to the environmental stressor and (optionally) the corresponding lower and upper bound of the 95% confidence interval. See Details for more info.
`cutoff_central`, `cutoff_lower`, `cutoff_upper`	`Numeric` `value` specifying the **exposure cut-off value** and (optionally) the corresponding lower and upper 95% confidence interval bounds. Default: 0. See Details for more info.

`pop_exp`	`Numeric vector` specifying the absolute size of the **population(s) exposed** to each exposure category. See Details for more info. *Only applicable in AR pathways; always required.*
`erf_eq_central`, `erf_eq_lower`, `erf_eq_upper`	`String` or `function` specifying the **exposure-response function** and (optionally) the corresponding lower and upper 95% confidence interval functions. See Details for more info. *Required in AR pathways; in RR pathways required only if* `rr_...` *argument(s) not specified.*
`rr_central`, `rr_lower`, `rr_upper`	`Numeric value` specifying the **central relative risk** estimate and (optionally) the corresponding lower and upper 95% confidence interval bounds. *Only applicable in RR pathways; not required if* `erf_eq_...` *argument(s) already specified.*
`rr_increment`	`Numeric value` specifying the **exposure increment** for which the provided relative risk is valid. See Details for more info. *Only applicable in RR pathways; not required if* `erf_eq_...` *argument(s) already specified.*
`erf_shape`	`String value` specifying the **exposure-response function shape** to be assumed. Options (no default): `"linear"`, `log_linear"`, `"linear_log"`, `"log_log"`. *Only applicable in RR pathways; not required if* `erf_eq_...` *argument(s) already specified.*
`bhd_central`, `bhd_lower`, `bhd_upper`	`Numeric value` or `numeric vector` providing the **baseline health data** of the health outcome of interest in the study population and (optionally) the corresponding lower bound and the upper 95% confidence interval bounds. See Details for more info. *Only applicable in RR pathways; always required.*
`prop_pop_exp`	`Numeric value` or `numeric vector` specifying the **population fraction(s) exposed** for each exposure (category). Default: 1. See Details for more info. *Only applicable in RR pathways.*
`geo_id_micro`, `geo_id_macro`	`Numeric vector` or `string vector` providing **unique IDs of the geographic area** considered in the assessment (`geo_id_micro`) and (optionally) providing higher-level IDs (`geo_id_macro`) to aggregate the geographic areas at. See Details for more info. *Only applicable in assessments with multiple geographic units.*
`age_group`	`Numeric vector` or `string vector` providing the **age groups** considered in the assessment. In case of use in `attribute_lifetable)()`, it must be a `numeric` and contain single year age groups. See Details for more info. *Optional argument for* `attribute_health()`*; needed for* `attribute_lifetable()`.
`sex`	`Numeric vector` or `string vector` specifying the **sex** of the groups considered in the assessment.*Optional argument.*
`dw_central`, `dw_lower`, `dw_upper`	`Numeric value` or `numeric vector` providing the **disability weight** associated with the morbidity health outcome of interest and (optionally) the corresponding lower bound and the upper 95% confidence interval bounds. *Only applicable in assessments of YLD (years lived with disability).*

`duration_central`, `duration_lower`, `duration_upper`	`Numeric value` or `numeric vector` providing the **duration** associated with the morbidity health outcome of interest in years and (optionally) the corresponding lower and upper bounds of the 95% confidence interval. Default: 1. See Details for more info. *Only applicable in assessments of YLD (years lived with disability).*
`info`	`String`, `data frame` or `tibble` providing **information about the assessment**. See Details for more info. *Optional argument.*
`population`	`Numeric vector For` `attribute_lifetable()`, it is an *obligatory argument* specifying the **mid-year populations** per age (i.e. age group size = 1 year) for the (first) year of analysis. `For` `attribute_health()` it is an *optional argument* which specifies the **population used to calculate attributable impacts rate** per 100 000 population. See Details for more info.

Details

What you put in is what you get out

The health metric you put in (e.g. absolute disease cases, deaths per 100 000 population, DALYs, prevalence, incidence, ...) is the one you get out.

Exception: if you enter a disability weight (via the `dw_...` arguments) the attributable impact will be in YLD.

Function arguments

`exp_central`, `exp_lower`, `exp_upper` In case of exposure bands enter only one exposure value per band (e.g. the means of the lower and upper bounds of the exposure bands).

`cutoff_central`, `cutoff_lower`, `cutoff_upper` The cutoff level refers to the exposure level below which no health effects occur in the same unit as the exposure. If exposure categories are used, the length of this input must be the same as in the `exp_...` argument(s).

`pop_exp` *Only applicable in AR pathways; always required.* In AR pathways the population exposed per exposure category is multiplied with the corresonding category-specific risk to obtain the absolute number of people affected by the health outcome.

`erf_eq_central`, `erf_eq_lower`, `erf_eq_upper` *Required in AR pathways; in RR pathways required only if rr_... arguments not specified.* Enter the exposure-response function as a `function`, e.g. output from `stats::splinefun()` or `stats::approxfun()`, or as a `string` formula, e.g. `"3+c+c^2"` (with the *c* representing the concentration/exposure). If you have x-axis (exposure) and y-axis (relative risk) value pairs of multiple points lying on the the exposure-response function, you could use e.g. `stats::splinefun(x, y, method="natural")` to derive the exposure-response function (in this example using a cubic spline natural interpolation). `rr_increment` *Only applicable in RR pathways.* Relative risks from the literature are valid for a specific increment in the exposure, in case of air pollution often 10 or 5 $\mu g/m^3$).

`bhd_central`, `bhd_lower`, `bhd_upper` *Only applicable in RR pathways.* Baseline health data for each exposure category must be entered.

`prop_pop_exp` *Only applicable in RR pathways.* In RR pathways indicates the fraction(s) (value(s) from 0 until and including 1) of the total population exposed to the exposure categories. See equation of the population attributable fraction for categorical exposure below.

`geo_id_macro`, `geo_id_micro` *Only applicable in assessments with multiple geographic units.* For example, if you provide the names of the municipalities under analysis to `geo_id_micro`, you might provide to `geo_id_macro` the corresponding region / canton / province names. Consequently, the vectors fed to `geo_id_micro` and `geo_id_macro` must be of the same length.

`age_group` Can be either `numeric` or `character`. If it is numeric, it refers to the first age of the age group. E.g. `c(0, 40, 80)` means age groups `[0, 40)`, `[40, 80)`, `>=80]`.

`info` *Optional argument.* Information entered to this argument will be added as column(s) names `info_1`, `info_2`, `info_...` to the results table. These additional columns can be used to further stratify the analysis in a secondary step (see example below).

`population` *Optional argument.* The population entered here is used to determine impact rate per 100 000 population. Note the requirement for the vector length in the paragraph *Assessment of multiple geographic units* below.

`duration_central`, `duration_lower`, `duration_upper` *Only applicable in assessments of YLD (years lived with disability).* Measured in years. A value of 1 (year) refers to the prevalence-based approach, while values above 1 to the incidence-based approach.

Methodology

This function can quantify the attributable health impacts by means of a relative risk or an absolute risk (depending on the health outcome).

- Relative risk: The comparative risk assessment approach (Murray et al. 2003) is applied by obtaining the population attributable fraction (percent of cases that are attributable to the exposure) based on the relative risk (WHO 2003).
- Absolute risk: The attributable cases are directly derived from population exposed (WHO 2011).

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- Relative risk
- Absolute risk

Value

This function returns a `list` containing:

1) `health_main` (`tibble`) containing the main results;

- `impact` (`numeric` column) attributable health burden/impact
- `pop_fraction` (`numeric` column) population attributable fraction; only applicable in relative risk assessments
- And many more

2) `health_detailed` (`list`) containing detailed (and interim) results.

- `input_args` (`list`) containing all the argument inputs used in the background
- `input_table` (`tibble`) containing the inputs after preparation
- `results_raw` (`tibble`) containing results for all combinations of input (geo units, uncertainty, age and sex specific data...)
- `results_by_...` (`tibble`) containing results stratified by each geographic unit, age or sex.

Author(s)

Alberto Castro & Axel Luyten

References

Murray CJ, Ezzati M, Lopez AD, Rodgers A, Hoorn SV (2003). “Comparative risk assessment: conceptual framework and design.” *Epidemiology*, **14**(4), 447-458. doi:10.1186/1478795411.

WHO (2003). “Introduction and methods: Assessing the environmental burden of disease at national and local levels.” World Health Organization. `https://www.who.int/publications/i/item/9241546204`.

WHO (2011). “Burden of disease from environmental noise: Quantification of healthy life years lost in Europe.” World Health Organization. `https://www.who.int/publications/i/item/burden-of-disease-from-environmental-noise-quantification-of-healthy-life-years-lost-in-europe`.

See Also

- Upstream: `prepare_exposure` (only if no exposure data available)
- Alternative: `attribute_lifetable`, `get_paf`, `get_risk`
- Downstream: `attribute_mod`, `compare`, `daly`, `multiexpose`, `standardize`, `monetize`, `socialize`

Examples

```
# Goal: attribute lung cancer cases to population-weighted PM2.5 exposure
# using relative risk

results <- attribute_health(
  erf_shape = "log_linear",
  rr_central = 1.369,             # Central relative risk estimate
  rr_increment = 10,              # per \mu g / m^3 increase in PM2.5 exposure
  exp_central = 8.85,             # Central exposure estimate in \mu g / m^3
  cutoff_central = 5,             # \mu g / m^3
  bhd_central = 30747             # Baseline health data: lung cancer incidence
 )

results$health_main$impact_rounded # Attributable cases


# Goal: attribute cases of high annoyance to (road traffic) noise exposure
# using absolute risk

results <- attribute_health(
  approach_risk = "absolute_risk",
  exp_central = c(57.5, 62.5, 67.5, 72.5, 77.5),
  pop_exp = c(387500, 286000, 191800, 72200, 7700),
  erf_eq_central = "78.9270-3.1162*c+0.0342*c^2"
)

results$health_main$impact_rounded # Attributable high annoyance cases
```

```
# Goal: attribute disease cases to PM2.5 exposure in multiple geographic
# units, such as municipalities, provinces, countries, ...

results <- attribute_health(
  geo_id_micro = c("Zurich", "Basel", "Geneva", "Ticino"),
  geo_id_macro = c("Ger","Ger","Fra","Ita"),
  rr_central = 1.369,
  rr_increment = 10,
  cutoff_central = 5,
  erf_shape = "log_linear",
  exp_central = c(11, 11, 10, 8),
  bhd_central = c(4000, 2500, 3000, 1500)
)

# Attributable cases (aggregated)
results$health_main$impact_rounded

# Attributable cases (disaggregated)
results$health_detailed$results_raw |> dplyr::select(
  geo_id_micro,
  geo_id_macro,
  impact_rounded
)


# Goal: determine attributable YLD (years lived with disability)
results  <- attribute_health(
  exp_central = 8.85,
  prop_pop_exp = 1,
  cutoff_central = 5,
  bhd_central = 1000,
  rr_central = 1.1,
  rr_increment = 10,
  erf_shape = "log_linear",
  duration_central = 100,
  dw_central = 1,
  info = "pm2.5_yld"
)

results$health_main$impact
```

`attribute_lifetable` *Attribute premature deaths or YLL to an environmental stressor using a life table approach*

Description

This function assesses premature deaths or years of life lost (YLL) attributable to exposure to an environmental stressor using a life table approach.

Usage

```
attribute_lifetable(
  age_group,
  sex,
  bhd_central,
  bhd_lower = NULL,
  bhd_upper = NULL,
  population,
  health_outcome = NULL,
  min_age = NULL,
  max_age = NULL,
  approach_exposure = "single_year",
  approach_newborns = "without_newborns",
  year_of_analysis,
  time_horizon = NULL,
  exp_central = NULL,
  exp_lower = NULL,
  exp_upper = NULL,
  cutoff_central = 0,
  cutoff_lower = NULL,
  cutoff_upper = NULL,
  erf_eq_central = NULL,
  erf_eq_lower = NULL,
  erf_eq_upper = NULL,
  rr_central = NULL,
  rr_lower = NULL,
  rr_upper = NULL,
  rr_increment = NULL,
  erf_shape = NULL,
  prop_pop_exp = 1,
  geo_id_micro = "a",
  geo_id_macro = NULL,
  info = NULL
)
```

Arguments

`age_group`	`Numeric vector` or `string vector` providing the **age groups** considered in the assessment. In case of use in `attribute_lifetable)()`, it must be a `numeric` and contain single year age groups. See Details for more info. *Optional argument for* `attribute_health()`*; needed for* `attribute_lifetable()`.
`sex`	`Numeric vector` or `string vector` specifying the **sex** of the groups considered in the assessment.*Optional argument.*

`bhd_central`, `bhd_lower`, `bhd_upper`	`Numeric value` or `numeric vector` providing the **baseline health data** of the health outcome of interest in the study population and (optionally) the corresponding lower bound and the upper 95% confidence interval bounds. See Details for more info. *Only applicable in RR pathways; always required.*
`population`	`Numeric vector For attribute_lifetable()`, it is an *obligatory argument* specifying the **mid-year populations** per age (i.e. age group size = 1 year) for the (first) year of analysis. `For attribute_health()` it is an *optional argument* which specifies the **population used to calculate attributable impacts rate** per 100 000 population. See Details for more info.
`health_outcome`	`String` specifying the desired result of the life table assessment. Options: `"deaths"` (premature deaths), `"yll"` (years of life lost).
`min_age`, `max_age`	`Numberic value` specifying the minimum and maximum age for which the exposure will affect the exposed population, respectively. Default `min_age`: 30. Default `max_age`: none. See Details for more info.
`approach_exposure`	`String` specifying whether exposure is constant or only in one year. Options: `"single_year"` (default), `"constant"`.
`approach_newborns`	`String` specifying whether newborns are to be considered in the years after the year of analysis or not. Options: `"without_newborns"` (default), `"with_newborns"`. See Details for more info.
`year_of_analysis`	`Numeric value` providing the first with exposure to the environmental stressor.
`time_horizon`	`Numeric value` specifying the time horizon (number of years) for which the attributable YLL or premature deaths are to be considered. See Details for more info. *Optional argument.*
`exp_central`, `exp_lower`, `exp_upper`	`Numeric value` or `numeric vector` specifying the **exposure level(s)** to the environmental stressor and (optionally) the corresponding lower and upper bound of the 95% confidence interval. See Details for more info.
`cutoff_central`, `cutoff_lower`, `cutoff_upper`	`Numeric value` specifying the **exposure cut-off value** and (optionally) the corresponding lower and upper 95% confidence interval bounds. Default: 0. See Details for more info.
`erf_eq_central`, `erf_eq_lower`, `erf_eq_upper`	`String` or `function` specifying the **exposure-response function** and (optionally) the corresponding lower and upper 95% confidence interval functions. See Details for more info. *Required in AR pathways; in RR pathways required only if* `rr_...` *argument(s) not specified.*
`rr_central`, `rr_lower`, `rr_upper`	`Numeric value` specifying the **central relative risk** estimate and (optionally) the corresponding lower and upper 95% confidence interval bounds. *Only applicable in RR pathways; not required if* `erf_eq_...` *argument(s) already specified.*

`rr_increment`	`Numeric value` specifying the **exposure increment** for which the provided relative risk is valid. See Details for more info. *Only applicable in RR pathways; not required if* `erf_eq_...` *argument(s) already specified.*
`erf_shape`	`String value` specifying the **exposure-response function shape** to be assumed. Options (no default): `"linear"`, `log_linear"`, `"linear_log"`, `"log_log"`. *Only applicable in RR pathways; not required if* `erf_eq_...` *argument(s) already specified.*
`prop_pop_exp`	`Numeric value` or `numeric vector` specifying the **population fraction(s) exposed** for each exposure (category). Default: 1. See Details for more info. *Only applicable in RR pathways.*
`geo_id_micro`, `geo_id_macro`	`Numeric vector` or `string vector` providing **unique IDs of the geographic area** considered in the assessment (`geo_id_micro`) and (optionally) providing higher-level IDs (`geo_id_macro`) to aggregate the geographic areas at. See Details for more info. *Only applicable in assessments with multiple geographic units.*
`info`	`String`, `data frame` or `tibble` providing **information about the assessment**. See Details for more info. *Optional argument.*

Details

Function arguments

`age_group` The numeric values must refer to 1 year age groups, e.g. `c(0:99)`. To convert multi-year/larger age groups to 1 year age groups use the function `prepare_lifetable()` (see its function documentation for more info).

`bhd_central,bhd_lower,bhd_upper` Deaths per age must be inputted with 1 value per age (i.e. age group size = 1 year). There must be greater than or equal to 1 deaths per age to avoid issues during the calculation of survival probabilities. If zeros show up in the last ages (e.g. age 98 = 0 deaths, 99 years old = 1), please sum the values and condensate last category (e.g. age 98 = 1).

`population` The population data must be inputted with 1 value per age (i.e. age group size = 1 year). The values must be greater than or equal to 1 per age to avoid issues during the calculation of survival probabilities. Mid-year population of year x can be approximated as the mean of either end-year populations of years x-1 and x or start-of-year populations of years x and x+1. For each age, the inputted values must be greater than or equal to 1 to avoid issues during the calculation of survival probabilities.

`approach_newborns` If `"with_newborns"` is selected, it is assumed that for each year after the year of analysis n babies (population aged 0) are born.

`time_horizon` Applicable for the following cases:

- YLL: `single_year` or `constant` exposure
- premature deaths: `constant` exposure

For example, if 10 is entered one is interested in the impacts of exposure during the year of analysis and the next 9 years (= 10 years in total). Default value: length of the numeric vector specified in the `age_group` argument.

`min_age`, `max_age` The `min_age` default value 30 implies that all adults aged 30 or older will be affected by the exposure; `max_age` analogeously specifies the age above which no health effects of the exposure are considered.

Methodology

The life table approach to obtain YLL and deaths requires population and baseline mortality data to be stratified by one year age groups. This function applies the same approach as the on applied in the WHO tool AirQ+ (WHO 2020), which is described in previous literature (Miller and Hurley 2003).

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- YLL and deaths with life table

Conversion of multi-year to single year age groups

To convert multi-year/larger age groups to 1 year age groups, use the `healthiar` function `prepare_lifetable()`.

Value

This function returns a `list` containing:

1) `health_main` (`tibble`) containing the main results;

- `impact` (`numeric` column) attributable health burden/impact
- `pop_fraction` (`numeric` column) population attributable fraction; only applicable in relative risk assessments
- And many more

2) `health_detailed` (`list`) containing detailed (and interim) results.

- `input_args` (`list`) containing all the argument inputs used in the background
- `input_table` (`tibble`) containing the inputs after preparation
- `results_raw` (`tibble`) containing results for all combinations of input (geo units, uncertainty, age and sex specific data...)
- `results_by_...` (`tibble`) containing results stratified by each geographic unit, age or sex.

Note

For this specific function, the return object `health_detailed` also contains `intermediate_calculations`. This is a nested `tibble` containing intermediate results, such as population projections and impact by age/year.

Author(s)

Alberto Castro & Axel Luyten

References

Miller BG, Hurley JF (2003). "Life table methods for quantitative impact assessments in chronic mortality." *Journal of Epidemiology and Community Health*, **57**(3), 200–206. ISSN 0143-005X, doi:10.1136/jech.57.3.200.

WHO (2020). "Health impact assessment of air pollution: AirQ+ life table manual." World Health Organization - Regional Office for Europe. `https://www.who.int/europe/publications/i/item/WHO-EURO-2020-1559-41310-56212`.

See Also

- Upstream: `prepare_exposure` (only if no exposure data available)
- Alternative: `attribute_health`, `get_paf`, `get_risk`
- Downstream: `attribute_mod`, `compare`, `daly`, `multiexpose`, `standardize`, `monetize`, `socialize`

Examples

```
# Goal: determine YLL attributable to air pollution exposure during one year
# using the life table approach
results <- attribute_lifetable(
  health_outcome = "yll",
  approach_exposure = "single_year",
  approach_newborns = "without_newborns",
  exp_central = 8.85,
  prop_pop_exp = 1,
  cutoff_central = 5,
  rr_central =  1.118,
  rr_increment = 10,
  erf_shape = "log_linear",
  age_group = exdat_lifetable$age_group,
  sex = exdat_lifetable$sex,
  bhd_central = exdat_lifetable$deaths,
  population = exdat_lifetable$midyear_population,
  year_of_analysis = 2019,
  min_age = 20
)
results$health_main$impact # Attributable YLL


# Goal: determine attributable premature deaths due to air pollution exposure
# during one year using the life table approach
results_pm_deaths <- attribute_lifetable(
  health_outcome = "deaths",
  approach_exposure = "single_year",
  exp_central = 8.85,
  prop_pop_exp = 1,
  cutoff_central = 5,
  rr_central =  1.118,
  rr_increment = 10,
  erf_shape = "log_linear",
  age_group = exdat_lifetable$age_group,
```

```
  sex = exdat_lifetable$sex,
  bhd_central = exdat_lifetable$deaths,
  population = exdat_lifetable$midyear_population,
  year_of_analysis = 2019,
  min_age = 20
)
results_pm_deaths$health_main$impact # Attributable premature deaths


# Goal: determine YLL attributable to air pollution exposure (exposure distribution)
# during one year using the life table approach
results <- attribute_lifetable(
  health_outcome = "yll",
  exp_central = rep(c(8, 9, 10), each = 100*2), # each = length of sex or age_group vector
  prop_pop_exp = rep(c(0.2, 0.3, 0.5), each = 100*2), # each = length of sex or age_group vector
  cutoff_central = 5,
  rr_central = 1.118,
  rr_lower = 1.06,
  rr_upper = 1.179,
  rr_increment = 10,
  erf_shape = "log_linear",
  age_group = rep(
    exdat_lifetable$age_group,
    times = 3), # times = number of exposure categories
  sex = rep(
    exdat_lifetable$sex,
    times = 3), # times = number of exposure categories
  population = rep(
    exdat_lifetable$midyear_population,
    times = 3), # times = number of exposure categories
  bhd_central = rep(
    exdat_lifetable$deaths,
    times = 3), # times = number of exposure categories
  year_of_analysis = 2019,
  min_age = 20
)
results$health_main$impact_rounded # Attributable YLL
```

`attribute_mod` *Create a scenario 2 by modifying an existing scenario 1 and determine attributable health impacts in it*

Description

This function quantifies the attributable health impacts in a new scenario 2 as follows:

- taking the input data of an existing scenario 1 (obtained using `attribute_health()`),
- modifying some of these input data of scenario 1 for the scenario 2 and
- calling in the background `attribute_health()` with the new data for scenario 2

Usage

```
attribute_mod(
  output_attribute,
  erf_shape = NULL,
  rr_central = NULL,
  rr_lower = NULL,
  rr_upper = NULL,
  rr_increment = NULL,
  erf_eq_central = NULL,
  erf_eq_lower = NULL,
  erf_eq_upper = NULL,
  exp_central = NULL,
  exp_lower = NULL,
  exp_upper = NULL,
  prop_pop_exp = NULL,
  pop_exp = NULL,
  cutoff_central = NULL,
  cutoff_lower = NULL,
  cutoff_upper = NULL,
  bhd_central = NULL,
  bhd_lower = NULL,
  bhd_upper = NULL,
  geo_id_micro = NULL,
  geo_id_macro = NULL,
  age_group = NULL,
  sex = NULL,
  population = NULL,
  info = NULL,
  min_age = NULL,
  max_age = NULL,
  approach_exposure = NULL,
  approach_newborns = NULL,
  year_of_analysis = NULL
)
```

Arguments

`output_attribute`	`List` containing the output of the function attribute() for scenario 1.
`erf_shape`	`String` value specifying the **exposure-response function shape** to be assumed. Options (no default): `"linear"`, `log_linear"`, `"linear_log"`, `"log_log"`. *Only applicable in RR pathways; not required if* `erf_eq_...` *argument(s) already specified.*
`rr_central`, `rr_lower`, `rr_upper`	`Numeric` value specifying the **central relative risk** estimate and (optionally) the corresponding lower and upper 95% confidence interval bounds. *Only applicable in RR pathways; not required if* `erf_eq_...` *argument(s) already specified.*

`rr_increment`	`Numeric value` specifying the **exposure increment** for which the provided relative risk is valid. See Details for more info. *Only applicable in RR pathways; not required if* `erf_eq_...` *argument(s) already specified.*
`erf_eq_central`, `erf_eq_lower`, `erf_eq_upper`	`String` or `function` specifying the **exposure-response function** and (optionally) the corresponding lower and upper 95% confidence interval functions. See Details for more info. *Required in AR pathways; in RR pathways required only if* `rr_...` *argument(s) not specified.*
`exp_central`, `exp_lower`, `exp_upper`	`Numeric value` or `numeric vector` specifying the **exposure level(s)** to the environmental stressor and (optionally) the corresponding lower and upper bound of the 95% confidence interval. See Details for more info.
`prop_pop_exp`	`Numeric value` or `numeric vector` specifying the **population fraction(s) exposed** for each exposure (category). Default: 1. See Details for more info. *Only applicable in RR pathways.*
`pop_exp`	`Numeric vector` specifying the absolute size of the **population(s) exposed** to each exposure category. See Details for more info. *Only applicable in AR pathways; always required.*
`cutoff_central`, `cutoff_lower`, `cutoff_upper`	`Numeric value` specifying the **exposure cut-off value** and (optionally) the corresponding lower and upper 95% confidence interval bounds. Default: 0. See Details for more info.
`bhd_central`, `bhd_lower`, `bhd_upper`	`Numeric value` or `numeric vector` providing the **baseline health data** of the health outcome of interest in the study population and (optionally) the corresponding lower bound and the upper 95% confidence interval bounds. See Details for more info. *Only applicable in RR pathways; always required.*
`geo_id_micro`, `geo_id_macro`	`Numeric vector` or `string vector` providing **unique IDs of the geographic area** considered in the assessment (`geo_id_micro`) and (optionally) providing higher-level IDs (`geo_id_macro`) to aggregate the geographic areas at. See Details for more info. *Only applicable in assessments with multiple geographic units.*
`age_group`	`Numeric vector` or `string vector` providing the **age groups** considered in the assessment. In case of use in `attribute_lifetable)()`, it must be a `numeric` and contain single year age groups. See Details for more info. *Optional argument for* `attribute_health()`*; needed for* `attribute_lifetable()`.
`sex`	`Numeric vector` or `string vector` specifying the **sex** of the groups considered in the assessment.*Optional argument.*
`population`	`Numeric vector For` `attribute_lifetable()`, it is an *obligatory argument* specifying the **mid-year populations** per age (i.e. age group size = 1 year) for the (first) year of analysis. `For` `attribute_health()` it is an *optional argument* which specifies the **population used to calculate attributable impacts rate** per 100 000 population. See Details for more info.
`info`	`String`, `data frame` or `tibble` providing **information about the assessment**. See Details for more info. *Optional argument.*

`min_age`, `max_age`
: `Numberic value` specifying the minimum and maximum age for which the exposure will affect the exposed population, respectively. Default `min_age`: 30. Default `max_age`: none. See Details for more info.

`approach_exposure`
: `String` specifying whether exposure is constant or only in one year. Options: `"single_year"` (default), `"constant"`.

`approach_newborns`
: `String` specifying whether newborns are to be considered in the years after the year of analysis or not. Options: `"without_newborns"` (default), `"with_newborns"`. See Details for more info.

`year_of_analysis`
: `Numeric value` providing the first with exposure to the environmental stressor.

Details

Methodology

This function calls in the background `attribute_health()`, which can use the relative risk (Murray et al. 2003) and absolute risk (WHO 2011).

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- Modification of scenarios
- Relative risk
- Absolute risk

Value

This function returns a `list` containing:

1) `health_main` (`tibble`) containing the main results;

- `impact` (`numeric` column) attributable health burden/impact
- `pop_fraction` (`numeric` column) population attributable fraction; only applicable in relative risk assessments
- And many more

2) `health_detailed` (`list`) containing detailed (and interim) results.

- `input_args` (`list`) containing all the argument inputs used in the background
- `input_table` (`tibble`) containing the inputs after preparation
- `results_raw` (`tibble`) containing results for all combinations of input (geo units, uncertainty, age and sex specific data...)
- `results_by_...` (`tibble`) containing results stratified by each geographic unit, age or sex.

Author(s)

Alberto Castro & Axel Luyten

References

Murray CJ, Ezzati M, Lopez AD, Rodgers A, Hoorn SV (2003). “Comparative risk assessment: conceptual framework and design.” *Epidemiology*, **14**(4), 447-458. doi:10.1186/1478795411.

WHO (2011). “Burden of disease from environmental noise: Quantification of healthy life years lost in Europe.” World Health Organization. `https://www.who.int/publications/i/item/burden-of-disease-from-environmental-noise-quantification-of-healthy-life-years-lost-in-europe`.

See Also

- Upstream: `attribute_health`, `attribute_lifetable`
- Downstream: `compare`

Examples

```
# Goal: adjust an existing healthiar scenario and determine the health
# impacts in the modified scenario

## First create a scenario to be modified
scenario_A <- attribute_health(
  exp_central = 8.85,   # EXPOSURE 1
  cutoff_central = 5,
  bhd_central = 25000,
  approach_risk = "relative_risk",
  erf_shape = "log_linear",
  rr_central = 1.118,
  rr_increment = 10
)

scenario_A$health_main$impact # Attributable impact in scenario A

## Modify scenario (adjust exposure value)
scenario_B <- attribute_mod(
  output_attribute = scenario_A,
  exp_central = 6       # EXPOSURE 2
)

scenario_B$health_main$impact # Attributable impact in scenario B
```

---

cba *Cost-benefit analysis*

Description

This function performs a cost-benefit analysis. It is assumed that the benefit is caused by the positive health impacts of a policy intervention, which generates some costs.

Usage

```
cba(
  output_attribute = NULL,
  impact_benefit = NULL,
  valuation,
  cost,
  discount_rate_benefit = NULL,
  discount_rate_cost = NULL,
  inflation_rate = NULL,
  real_growth_rate = NULL,
  discount_shape = "exponential",
  n_years_benefit = 1,
  n_years_cost = 0
)
```

Arguments

`output_attribute`	`List` produced by `healthiar::attribute_health()`, `healthiar::attribute_lifetable()` or `healthiar::compare()` as results.
`impact_benefit`	`Numeric value` referring to the positive health impact as result of a reduction of harmful exposure.
`valuation`	`Numberic value` referring to unit value of a health impact.
`cost`	`Numeric value` referring to the investment cost to achieve the reduction of exposure.
`discount_rate_benefit`, `discount_rate_cost`	`Numeric value` referring to the the discount rate used in the benefit and the cost side (respectively). Their values determine the approach of cost-benefit analysis: direct approach (if the same discount_rate is used for cost and benefit) and indirect approach (different discount rates).
`inflation_rate`	`Numeric value` between 0 and 1 referring to the annual inflation (increase of prices). This value is used to adjust monetization for inflation (converting nominal into real values by appyling a deflator). If this adjustment for inflation is not needed leave this argument empty (default value = NULL).
`real_growth_rate`	`Numeric value` between 0 and 1 referring to the annual real-term appreciation in the societal value of health (e.g., income elasticity). This adjusts the valuation upward to reflect rising wealth, independent of general price inflation.
`discount_shape`	`String` referring to the assumed equation for the discount factor. By default: `"exponential"`. Otherwise: `"hyperbolic_harvey_1986"` or `"hyperbolic_mazur_1987"`.
`n_years_benefit`, `n_years_cost`	`Numeric value` referring to number of years in the future to be considered in the benefit and cost side (respectively). Years for discounting and/or inflation. Be aware that the year 0 (without discounting/inflation, i.e. the present) is not be counted here. If a vector is entered in the argument impact, n_years does not need to be entered (length of impact = n_years + 1)

Details

Methodology

This function provides as output three types of cost-benefit indicators (Boardman et al. 2018):

- net benefit
- cost-benefit ratio
- return of investment

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- cost-benefit analysis
- monetization

Value

This function returns a `list` containing:

1) `cba_main` (`tibble`) containing the main CBA results;

- `net_benefit` (`numeric` column) containing the difference between benefit and cost (i.e. benefit - cost)
- `benefit` (`numeric` column) containing discounted benefit (i.e. monetized attributable health impact)
- `cost` (`numeric` column) containing discounted cost
- And many more

2) `cba_detailed` (`list`) containing detailed (and interim) results.

- `benefit` (`list`)
- `cost` (`tibble`)

If the argument `output_attribute` was specified, then the two results elements are added to the existing output.

Author(s)

Alberto Castro & Axel Luyten

References

Boardman AE, Greenberg DH, Vining AR, Weimer DL (2018). *Cost-Benefit Analysis: Concepts and Practice*, 5th edition. Cambridge University Press, Cambridge, UK. ISBN 978-1108401296.

See Also

- Upstream: `attribute_health`
- Alternative: `monetize`

Examples

```
# Goal: performs a cost-benefit analysis using an existing output
# of a attribute_... function

output_attribute <- attribute_health(
  erf_shape = "log_linear",
  rr_central = 1.369,
  rr_increment = 10,
  exp_central = 8.85,
  cutoff_central = 5,
  bhd_central = 30747
)

results <- cba(
  output_attribute = output_attribute,
  valuation = 50000,
  cost = 100000000,
  discount_shape = "exponential",
  discount_rate_benefit = 0.03,
  discount_rate_cost = 0.03,
  n_years_benefit = 5,
  n_years_cost = 5
)

results$cba_main |>
  dplyr::select(benefit, cost, net_benefit)
```

`compare` *Compare the attributable health impacts between two scenarios*

Description

This function calculates the health impacts between two scenarios (e.g. before and after a intervention in a health impact assessments) using either the delta or pif approach.

Usage

```
compare(
  output_attribute_scen_1,
  output_attribute_scen_2,
  approach_comparison = "delta"
)
```

Arguments

`output_attribute_scen_1`
Scenario 1 as in the output of attribute()

`output_attribute_scen_2`	Scenario 2 as in the output of attribute()
`approach_comparison`	`String` showing the method of comparison. Options: "delta" or "pif".

Details

Methodology This function compares the attributable health impacts in scenario 1 with scenario 2. It can use two approaches:

- Delta: Subtraction of health impacts in the two scenarios (two PAF) (WHO Regional Office for Europe 2014)
- Potential impact fraction (PIF): Single PIF for both scenarios (WHO 2003)

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- comparison of two health scenarios

Specifications of the comparison approach

Please, note that the PIF comparison approach assumes same baseline health data for scenario 1 and 2 (e.g. comparison of two scenarios at the same time point). With the delta comparison approach, the difference between two scenarios is obtained by subtraction. The delta approach is suited for all comparison cases, allowing a comparison of a situation now with a situation in the future.

IMPORTANT: If your aim is to quantify health impacts from a policy intervention, be aware that you should use the same year of analysis and therefore same health baseline data in both scenarios. The only variable that should change is the exposure (as a result of the intervention).

Comparing DALY

If you want to use `compare()` DALY with `daly()`, do not enter the output of `daly()` in `compare()`. Instead, follow these steps:

1) use `compare()` for YLL and YLD separately

2) use `daly()` inserting the output of both compare()

Alternatively, you can use `attribute_health` to quantify DALY entering DALY in the argument `bhd_central` and then use `compare()`

Value

This function returns a `list` containing:

1) `health_main` (`tibble`) containing the main results from the comparison;

- `impact` (`numeric` column) difference in attributable health burden/impact between scenario 1 and 2
- `impact_scen_1` (`numeric` column) attributable health impact of scenario 1
- `impact_scen_2` (`numeric` column) attributable health impact of scenario 2
- And many more

2) `health_detailed` (`list`) containing detailed (and interim) results from the comparison.

- `results_raw` (`tibble`) containing comparison results for each combination of input uncertainty for both scenario 1 and 2
- `results_by_geo_id_micro` (`tibble`) containing comparison results for each geographic unit under analysis (specified in `geo_id_micro` argument)
- `results_by_geo_id_macro` (`tibble`) containing comparison results for each aggregated geographic unit under analysis (specified in `geo_id_macro` argument))
- `input_table` (`list`) containing the inputs to each relevant argument for both scenario 1 and 2
- `input_args` (`list`) containing all the argument inputs for both scenario 1 and 2 used in the background
- `scen_1` (`tibble`) containing results for scenario 1
- `scen_2` (`tibble`) containing results for scenario 2

Author(s)

Alberto Castro & Axel Luyten

References

WHO Regional Office for Europe (2014). *WHO Expert Meeting: Methods and tools for assessing the health risks of air pollution at local, national and international level. Meeting report; 12-13 May 2014; Bonn, Germany*. WHO Regional Office for Europe, Copenhagen. https://iris.who.int/handle/10665/142940.

WHO (2003). “Introduction and methods: Assessing the environmental burden of disease at national and local levels.” World Health Organization. https://www.who.int/publications/i/item/9241546204.

See Also

- Upstream: `attribute_health`, `attribute_mod`, `standardize`,
- Downstream: `daly`

Examples

```
# Goal: comparison of two scenarios with delta approach
scenario_A <- attribute_health(
  exp_central = 8.85,   # EXPOSURE 1
  cutoff_central = 5,
  bhd_central = 25000,
  approach_risk = "relative_risk",
  erf_shape = "log_linear",
  rr_central = 1.118,
  rr_increment = 10
)
scenario_B <- attribute_health(
  exp_central = 6,     # EXPOSURE 2
  cutoff_central = 5,
```

```
    bhd_central = 25000,
    approach_risk = "relative_risk",
    erf_shape = "log_linear",
    rr_central = 1.118,
    rr_increment = 10
  )
  results <- compare(
  approach_comparison = "delta",
  output_attribute_scen_1 = scenario_A,
  output_attribute_scen_2 = scenario_B
  )
  # Inspect the difference, stored in the \code{impact} column
  results$health_main |>
    dplyr::select(impact, impact_scen_1, impact_scen_2) |>
    print()

  # Goal: comparison of two scenarios with potential impact fraction (pif) approach
  output_attribute_scen_1 <- attribute_health(
    exp_central = 8.85,   # EXPOSURE 1
    cutoff_central = 5,
    bhd_central = 25000,
    approach_risk = "relative_risk",
    erf_shape = "log_linear",
    rr_central = 1.118, rr_lower = 1.060, rr_upper = 1.179,
    rr_increment = 10
  )
  output_attribute_scen_2 <- attribute_health(
    exp_central = 6,      # EXPOSURE 2
    cutoff_central = 5,
    bhd_central = 25000,
    approach_risk = "relative_risk",
    erf_shape = "log_linear",
    rr_central = 1.118, rr_lower = 1.060, rr_upper = 1.179,
    rr_increment = 10
  )
  results <- compare(
    output_attribute_scen_1 = output_attribute_scen_1,
    output_attribute_scen_2 = output_attribute_scen_2,
    approach_comparison = "pif"
  )
  # Inspect the difference, stored in the impact column
  results$health_main$impact
```

`daly` *Attributable disability-adjusted life years*

Description

This function quantifies the attributable disability-adjusted life years (DALY) by summing the two DALY components: years of life lost (YLL) and years lived with disability (YLD).

Usage

```
daly(output_attribute_yll, output_attribute_yld)
```

Arguments

`output_attribute_yll`, `output_attribute_yld`	`variable` containing YLL or YLD results of a `attribute_...()` function call, respectively.

Details

Methodology

This function sums the years of life lost (YLL) and years lived with disability (YLD) to obtain the disability-adjusted lost years (DALY) (GBD 2019 Risk Factors Collaborators 2020).

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- dalys

Value

This function returns a `list` containing:

1) `health_main` (`tibble`) containing the main results;

- `impact` (`numeric` column) attributable health burden/impact in DALY
- `impact_yld` (`numeric` column) attributable health burden/impact in YLD
- `impact_yll` (`numeric` column) attributable health burden/impact in YLL
- `dw` (`numeric` column) disability weight used for YLD calculation
- And many more

2) `health_detailed` (`list`) containing detailed (and interim) results.

- `results_raw` (`tibble`) containing results for each combination of input uncertainty
- `results_by_geo_id_micro` (`tibble`) containing results for each geographic unit under analysis (specified in `geo_id_micro` argument)
- `input_args` (`list`) containing all the argument inputs used in the background

Author(s)

Alberto Castro & Axel Luyten

References

GBD 2019 Risk Factors Collaborators (2020). “Global burden of 87 risk factors in 204 countries and territories, 1990–2019.” *The Lancet*. doi:10.1016/S01406736(20)307522.

See Also

- Upstream: `attribute_health`, `attribute_lifetable`

Examples

```
# Goal: obtain DALY (disability-adjusted life years) from two existing \code{attribute_...} outputs
# Step 1: Create YLL (years of life lost) assessment
results_yll <- attribute_lifetable(
  health_outcome = "yll",
  approach_exposure = "single_year",
  approach_newborns = "without_newborns",
  exp_central = 8.85,
  prop_pop_exp = 1,
  cutoff_central = 5,
  rr_central =  1.118,
  rr_increment = 10,
  erf_shape = "log_linear",
  age_group = exdat_lifetable$age_group,
  sex = exdat_lifetable$sex,
  bhd_central = exdat_lifetable$deaths,
  population = exdat_lifetable$midyear_population,
  year_of_analysis = 2019,
  min_age = 20
)
# Step 2: Create YLD (years lived with disability) assessment
results_yld  <- attribute_health(
  exp_central = 8.85,
  prop_pop_exp = 1,
  cutoff_central = 5,
  bhd_central = 1000,
  rr_central = 1.1,
  rr_increment = 10,
  erf_shape = "log_linear",
  duration_central = 100,
  dw_central = 0.5,
  info = "pm2.5_yld"
)
# Step 3: obtain DALY
results <- daly(
  output_attribute_yll = results_yll,
  output_attribute_yld = results_yld
)
# Attributable impact in DALY
results$health_main |>
  dplyr::select(impact, impact_yll, impact_yld)
```

`discount` *Discount health impacts*

Description

This function calculates discounted health impacts (without valuation).

Usage

```
discount(
  output_attribute = NULL,
  impact = NULL,
  discount_rate = NULL,
  n_years = NULL,
  discount_shape = NULL
)
```

Arguments

`output_attribute`	`List` produced by `healthiar::attribute_health()`, `healthiar::attribute_lifetable()` or `healthiar::compare()` as results.
`impact`	`Numberic value` referring to the health impacts to be monetized (without attribute function). If a `Numberic vector` is entered multiple assessments (by year) will be carried out. Be aware that the value for year 0 (current) must be entered, while n_years does not include the year 0. Thus, length of impact = n_years + 1.
`discount_rate`	`Numeric value` showing the discount rate for future years.
`n_years`	`Numeric value` referring to number of years in the future to be considered in the discounting and/or inflation. Be aware that the year 0 (without discounting/inflation, i.e. the present) is not be counted here. If a vector is entered in the argument impact, n_years does not need to be entered (length of impact = n_years + 1).
`discount_shape`	`String` referring to the assumed equation for the discount factor. By default: `"exponential"`. Otherwise: `"hyperbolic_harvey_1986"` or `"hyperbolic_mazur_1987"`.

Details

Methodology

This function applies a discount to attributable health impacts into the future.

One of the following three discount shapes can be selected:

- Exponential (Frederick et al. 2002)
- Hyperbolic as Harvey (1986)
- Hyperbolic as Mazur (1987)

Burden of disease studies may be interested in calculating + discounted health impacts over time, and these may also be used in economic evaluation models, where benefits are not monetized. For this specific purpose, this function is offered.

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- Monetization

Value

This function returns a `list` containing:

1) `monetization_main` (`tibble`) containing the main monetized results;

- `monetized_impact` (`numeric` column)
- `discount_factor` (`numeric` column) calculated based on the entered `discount_rate`
- And many more

2) `monetization_detailed` (`list`) containing detailed (and interim) results.

- `results_by_year` (`tibble`)
- `health_raw` (`tibble`) containing the monetized results for each for each combination of input uncertainty that were provided to the initial `attribute_health()` call

If the argument `output_attribute` was specified, then the two results elements are added to the existing output.

Author(s)

Alberto Castro & Axel Luyten

References

Frederick S, Loewenstein G, O'Donoghue T (2002). "Time Discounting and Time Preference: A Critical Review." *Journal of Economic Literature*, **40**(2), 351–401. doi:10.1257/002205102320161311.

Harvey CM (1986). "Value Functions for Infinite-Period Planning." *Management Science*, **32**(9), 1123–1139. doi:10.1287/mnsc.32.9.1123.

Mazur JE (1987). "An adjusting procedure for studying delayed reinforcement." In Commons ML, Mazur JE, Nevin JA, Rachlin H (eds.), *Quantitative Analyses of Behavior: Volume V. The Effect of Delay and of Intervening Events on Reinforcement Value*, 55–73. Lawrence Erlbaum Associates, Hillsdale, NJ. ISBN 0-89859-800-1.

See Also

- Upstream: `attribute_health`, `attribute_health`
- Alternative: `monetize`

Examples

```
# Goal: discount attributable health impacts
results <- discount(
  impact = 20000,
  discount_shape = "exponential",
  discount_rate = 0.03,
  n_years = 20
)
results$monetization_main$monetized_impact
```

`get_discount_factor` *Get discount factor*

Description

This function calculates the discount factor based on discount rate.

Usage

```
get_discount_factor(discount_rate, n_years, discount_shape = "exponential")
```

Arguments

`discount_rate`	`Numeric` value showing the discount rate for future years.
`n_years`	`Numeric` value referring to number of years in the future to be considered in the discounting and/or inflation. Be aware that the year 0 (without discounting/inflation, i.e. the present) is not be counted here. If a vector is entered in the argument impact, n_years does not need to be entered (length of impact = n_years + 1).
`discount_shape`	`String` referring to the assumed equation for the discount factor. By default: `"exponential"`. Otherwise: `"hyperbolic_harvey_1986"` or `"hyperbolic_mazur_1987"`.

Details

Methodology

This function is called inside `monetize()`.

One of the following three discount shapes can be selected:

- Exponential (Frederick et al. 2002)
- Hyperbolic as Harvey (1986)
- Hyperbolic as Mazur (1987)

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- Monetization

Value

This function returns the `numeric` discount factor.

Author(s)

Alberto Castro & Axel Luyten

References

Frederick S, Loewenstein G, O'Donoghue T (2002). "Time Discounting and Time Preference: A Critical Review." *Journal of Economic Literature*, **40**(2), 351–401. doi:10.1257/002205102320161311.

Harvey CM (1986). "Value Functions for Infinite-Period Planning." *Management Science*, **32**(9), 1123–1139. doi:10.1287/mnsc.32.9.1123.

Mazur JE (1987). "An adjusting procedure for studying delayed reinforcement." In Commons ML, Mazur JE, Nevin JA, Rachlin H (eds.), *Quantitative Analyses of Behavior: Volume V. The Effect of Delay and of Intervening Events on Reinforcement Value*, 55–73. Lawrence Erlbaum Associates, Hillsdale, NJ. ISBN 0-89859-800-1.

See Also

- Alternative: `monetize`

Examples

```
get_discount_factor(
  discount_rate = 0.07,
  n_years = 5
 )
```

`get_inflation_factor` *Get inflation factor*

Description

This function calculates the inflation factor based on inflation rate.

Usage

```
get_inflation_factor(n_years, inflation_rate = NULL, is_deflation = FALSE)
```

Arguments

`n_years`	`Numeric value` referring to number of years in the future to be considered in the discounting and/or inflation. Be aware that the year 0 (without discounting/inflation, i.e. the present) is not be counted here. If a vector is entered in the argument impact, n_years does not need to be entered (length of impact = n_years + 1).
`inflation_rate`	`Numeric value` between 0 and 1 referring to the annual inflation (increase of prices). This value is used to adjust monetization for inflation (converting nominal into real values by appyling a deflator). If this adjustment for inflation is not needed leave this argument empty (default value = NULL).
`is_deflation`	`Boolean value` (TRUE vs. FALSE) referring to the type of inflation factor. FALSE (default) means inflate present values to future nominal values, while TRUE means deflate future nominal values to present real values

Details

Methodology

This function is called inside `monetize()`.

It calculates the inflation factor based on the inflation rate and the number of years into the future as described in Brealey et al. (2023).

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- Monetization

Value

This function returns the `numeric` inflation factor.

Author(s)

Alberto Castro & Axel Luyten

References

Brealey RA, Myers SC, Allen F, Benninga S, Read J (2023). *Principles of Corporate Finance*, 14th edition. McGraw-Hill Education, New York, NY. ISBN 978-1264117464.

See Also

- Alternative: `monetize`

Examples

```
get_inflation_factor(
  inflation_rate = 0.02,
  n_years = 5
)
```

`get_paf` *Get population attributable fraction*

Description

This function calculates the population attributable fraction (PAF) of a health outcome due to exposure to an environmental stressor

Usage

```
get_paf(rr_at_exp, prop_pop_exp)
```

Arguments

`rr_at_exp`	`Numerical value` Risk estimate of the concentration response function for a specific concentration. The population attributable fraction is normally calculated using the risk estimate that refers to the concentration that reflects the population exposure and the cut-off. This risk estimate is obtained after re-scaling from the epidemiological study with a particular increment (e.g. for PM2.5 10 or 5 ug/m3) to the aimed concentration.
`prop_pop_exp`	`Numeric value` or `numeric vector` specifying the **population fraction(s) exposed** for each exposure (category). Default: 1. See Details for more info. *Only applicable in RR pathways.*

Details

Methodology

This function is called internally inside other `healthiar` functions, e.g. `attribute_health()`. The function calculates the population attributable fraction (i.e. the percent of health cases that are attributable to the exposure) based on the relative risk as described in the extensive existing literature (WHO 2003).

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- relative risk

Value

This function returns the population attributable fraction as a `numeric value`.

Author(s)

Alberto Castro & Axel Luyten

References

WHO (2003). "Introduction and methods: Assessing the environmental burden of disease at national and local levels." World Health Organization. https://www.who.int/publications/i/item/9241546204.

See Also

- Alternative: `attribute_health`, `attribute_lifetable`

Examples

```
# Goal: calculate PAF based on RR and the proportion of population exposed
get_paf(rr = 1.062, prop_pop_exp = 1)
```

`get_pif` *Get potential impact fraction (PIF)*

Description

This function calculates the potential impact fraction of a health outcome due to exposure to an environmental stressor

Usage

```
get_pif(rr_at_exp_1, rr_at_exp_2, prop_pop_exp_1, prop_pop_exp_2)
```

Arguments

`rr_at_exp_1`	`Numerical value` showing the risk estimate of the concentration response function for a specific concentration in the scenario 1. The population attributable fraction is normally calculated using the risk estimate that refers to the concentration that reflects the population exposure and the cut-off. This risk estimate is obtained after re-scaling from the epidemiological study with a particular increment (e.g. for PM2.5 10 or 5 ug/m3) to the aimed concentration.
`rr_at_exp_2`	`Numerical value` showing the risk estimate of the concentration response function for a specific concentration in the scenario 2. The population attributable fraction is normally calculated using the risk estimate that refers to the concentration that reflects the population exposure and the cut-off. This risk estimate is obtained after re-scaling from the epidemiological study with a particular increment (e.g. for PM2.5 10 or 5 ug/m3) to the aimed concentration.
`prop_pop_exp_1`	`Numerical value` showing the fraction ([0,1]) of population exposed to the environmental stressor in the scenario 1. Per default = 1 (i.e. 100% of population is exposed).
`prop_pop_exp_2`	`Numerical value` showing the fraction ([0,1]) of population exposed to the environmental stressor in the scenario 1. Per default = 1 (i.e. 100% of population is exposed).

Details

Methodology

This function is internally used inside the `healthiar` function `compare()`. The potential impact fraction (PIF) is calculated as described in WHO (2003), Murray et al. (2003) and Askari and Namayandeh (2020).

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- comparison of two health scenarios

Value

This function returns the potential impact fraction as a `numeric value`.

Author(s)

Alberto Castro & Axel Luyten

References

Askari M, Namayandeh SM (2020). "The Difference between the Population Attributable Risk (PAR) and the Potentioal Impact Fraction (PIF)." *Iranian Journal of Public Health*, **49**(10), 2018–2019. doi:10.18502/ijph.v49i10.4713, `https://pmc.ncbi.nlm.nih.gov/articles/PMC7719653/`.

Murray CJL, Ezzati M, Lopez AD, Rodgers A, Vander Hoorn S (2003). "Comparative quantification of health risks conceptual framework and methodological issues." *Popul. Health Metr.*, **1**(1), 1.

WHO (2003). "Introduction and methods: Assessing the environmental burden of disease at national and local levels." World Health Organization. `https://www.who.int/publications/i/item/9241546204`.

See Also

- Alternative: `compare`

Examples

```
# Goal: calculate the potential impact fraction (PIF)
results <- get_pif(
  rr_at_exp_1 = 1.043879,
  rr_at_exp_2 = 1.011217,
  prop_pop_exp_1 = 1,
  prop_pop_exp_2 = 1
)
print(results)
```

`get_risk` *Get the relative risk of an exposure level*

Description

This function re-scales the relative risk from the increment value in the epidemiological study (e.g. for PM2.5 10 or 5 ug/m3) to the actual population exposure

Usage

```
get_risk(
  erf_shape = NULL,
  rr = NULL,
  rr_increment = NULL,
  erf_eq = NULL,
  cutoff = 0,
  exp
)
```

Arguments

`erf_shape`	`String value` specifying the **exposure-response function shape** to be assumed. Options (no default): `"linear"`, `log_linear"`, `"linear_log"`, `"log_log"`. *Only applicable in RR pathways; not required if* `erf_eq_...` *argument(s) already specified.*
`rr`	`Numeric value` or `numeric vector` specifying the **relative risk** estimate(s) and (optionally) the corresponding lower and upper 95% confidence interval bounds. Not required if the `erf_eq` argument is already specified.
`rr_increment`	`Numeric value` specifying the **exposure increment** for which the provided relative risk is valid. See Details for more info. *Only applicable in RR pathways; not required if* `erf_eq_...` *argument(s) already specified.*
`erf_eq`	`String` or `function` specifying the **exposure-response function** and (optionally) the corresponding lower and upper 95% confidence interval functions. See Details and Examples sections below.
`cutoff`	`Numeric value` specifying the **exposure cut-off value** (i.e. the exposure level below which no health effects occur) and (optionally) the corresponding lower and upper 95% confidence interval bounds.
`exp`	`Numeric value` or `numeric vector` specifying the **exposure level(s)** to the environmental stressor (e.g. annual population-weighted mean) and (optionally) the corresponding lower and upper bound of the 95% confidence interval.

Details

Function arguments `erf_eq` If the function is provided as `string`, it can only contain the variable c (exposure), e.g. "3+c+c^2". If the function is provided as a `function`, the object must be of the class function. If only the values of the x-axis (exposure) and y axis (relative risk) of the dots in the exposure-response function are available, a cubic spline natural interpolation can be assumed to get the function using, e.g., `stats::splinefun(x, y, method="natural")`

Methodology

This function is called internally inside other `healthiar` functions, e.g. `attribute_health()`. The function calculates the relative risk at the exposure level based on the relative risk available in the epidemiological literature and the assumed shape of the exposure-response function (Pozzer et al. 2023; Lehtomäki et al. 2025).

Detailed information about the methodology (including corresponding equations and literature) is available in the package vignette. More specifically, see chapters:

- relative risk

Value

This function returns the `numeric` risk value(s) at the specified exposure level(s), referred to as *rr_at_exp* in the relative risk equations above.

Author(s)

Alberto Castro & Axel Luyten

References

Lehtomäki H, Aasvang GM, Sulo G, Denby BR, Hänninen OO, Brauer M, Pereira G, Dadras O, Bølling AK (2025). "Burden of disease attributable to PM2.5 at low exposure levels: impact of methodological choices." *Environmental Health*, **25**(1), 4. ISSN 1476-069X, doi:10.1186/s12940-02501250y.

Pozzer A, Anenberg SC, Dey S, Haines A, Lelieveld J, Chowdhury S (2023). "Mortality Attributable to Ambient Air Pollution: A Review of Global Estimates." *GeoHealth*, **7**(1), e2022GH000711. doi:10.1029/2022GH000711, e2022GH000711 2022GH000711, https://agupubs.onlinelibrary.wiley.com/doi/pdf/10.1029/20 `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2022GH000711`.

See Also

- Alternative: `attribute_health`, `attribute_lifetable`

Examples

```
# Goal: scale relative risk to observed exposure level
get_risk(
  rr = 1.05,
  rr_increment = 10,
  erf_shape = "linear",
  exp = 10,
  cutoff = 5
)

# Goal: determine the absolute risk for high annoyance at specific noise exposure levels
get_risk(
```

```
  erf_eq = "78.9270-3.1162*c+0.0342*c^2",
  exp = c(57.5, 62.5, 67.5, 72.5, 77.5)
)

# Goal: attribute COPD cases to air pollution exposure
# by applying a user-defined exposure response function,
# e.g. MR-BRT curves from Global Burden of Disease study.
get_risk(
  erf_eq = splinefun(
    x = c(0, 5, 10, 15, 20, 25, 30, 50, 70, 90, 110),
    y = c(1.00, 1.04, 1.08, 1.12, 1.16, 1.20, 1.23, 1.35, 1.45, 1.53, 1.60),
    method = "natural"),
  exp = c(8, 9, 10)
)
```

`monetize` *Monetize health impacts*

Description

This function monetizes health impacts

Usage

```
monetize(
  output_attribute = NULL,
  impact = NULL,
  valuation,
  discount_rate = NULL,
  discount_shape = "exponential",
  n_years = NULL,
  inflation_rate = NULL,
  real_growth_rate = NULL,
  info = NULL
)
```

Arguments

`output_attribute`	`List` produced by `healthiar::attribute_health()`, `healthiar::attribute_lifetable()` or `healthiar::compare()` as results.
`impact`	`Numberic value` referring to the health impacts to be monetized (without attribute function). If a `Numberic vector` is entered multiple assessments (by year) will be carried out. Be aware that the value for year 0 (current) must be entered, while n_years does not include the year 0. Thus, length of impact = n_years + 1.

`valuation`	`Numberic value` referring to unit value of a health impact.
`discount_rate`	`Numeric value` showing the discount rate for future years.
`discount_shape`	`String` referring to the assumed equation for the discount factor. By default: `"exponential"`. Otherwise: `"hyperbolic_harvey_1986"` or `"hyperbolic_mazur_1987"`.
`n_years`	`Numeric value` referring to number of years in the future to be considered in the discounting and/or inflation. Be aware that the year 0 (without discounting/inflation, i.e. the present) is not be counted here. If a vector is entered in the argument impact, n_years does not need to be entered (length of impact = n_years + 1).
`inflation_rate`	`Numeric value` between 0 and 1 referring to the annual inflation (increase of prices). This value is used to adjust monetization for inflation (converting nominal into real values by appyling a deflator). If this adjustment for inflation is not needed leave this argument empty (default value = NULL).
`real_growth_rate`	`Numeric value` between 0 and 1 referring to the annual real-term appreciation in the societal value of health (e.g., income elasticity). This adjusts the valuation upward to reflect rising wealth, independent of general price inflation.
`info`	`String`, `data frame` or `tibble` providing **information about the assessment**. Only attached if `impact` is entered by the users. If `output_attribute` is entered, use `info` in that function or add the column manually. *Optional argument.*

Details

Methodology

This function monetize health impacts valuating them and applying discounting (Frederick et al. 2002) and considering inflation (Brealey et al. 2023).

If the monetized values require adjustment for inflation, a deflator based on `inflation_rate` can be applied (HM Treasury 2022).

If the monetized values require adjustment for base valuation upward, a factor based `valuation growth` can be applied (OECD 2012).

One of the following three discount shapes can be selected:

- Exponential (Frederick et al. 2002)
- Hyperbolic as Harvey (1986)
- Hyperbolic as Mazur (1987)

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- Monetization

Value

This function returns a `list` containing:

1) `monetization_main` (`tibble`) containing the main monetized results;

- `monetized_impact` (`numeric` column)

  - `discount_factor` (`numeric` column) calculated based on the entered `discount_rate`
  - And many more

2) `monetization_detailed` (`list`) containing detailed (and interim) results.

  - `results_by_year` (`tibble`)
  - `health_raw` (`tibble`) containing the monetized results for each for each combination of input uncertainty that were provided to the initial `attribute_health()` call

If the argument `output_attribute` was specified, then the two results elements are added to the existing output.

Author(s)

Alberto Castro & Axel Luyten

References

Brealey RA, Myers SC, Allen F, Benninga S, Read J (2023). *Principles of Corporate Finance*, 14th edition. McGraw-Hill Education, New York, NY. ISBN 978-1264117464.

Frederick S, Loewenstein G, O'Donoghue T (2002). "Time Discounting and Time Preference: A Critical Review." *Journal of Economic Literature*, **40**(2), 351–401. doi:10.1257/002205102320161311.

HM Treasury (2022). *The Green Book: Central Government Guidance on Appraisal and Evaluation*. HM Treasury, London, UK. `https://www.gov.uk/government/publications/the-green-book-appraisal-and-`

Harvey CM (1986). "Value Functions for Infinite-Period Planning." *Management Science*, **32**(9), 1123–1139. doi:10.1287/mnsc.32.9.1123.

Mazur JE (1987). "An adjusting procedure for studying delayed reinforcement." In Commons ML, Mazur JE, Nevin JA, Rachlin H (eds.), *Quantitative Analyses of Behavior: Volume V. The Effect of Delay and of Intervening Events on Reinforcement Value*, 55–73. Lawrence Erlbaum Associates, Hillsdale, NJ. ISBN 0-89859-800-1.

OECD (2012). *Mortality Risk Valuation in Environment, Health and Transport Policies*. OECD Publishing, Paris. doi:10.1787/9789264130807en.

See Also

  - Upstream: `attribute_health`, `attribute_lifetable`, `compare`
  - Alternative: `get_inflation_factor`, `get_discount_factor`, `cba`

Examples

```
# Goal: monetize the attributable impacts of an existing healthiar
# assessment
output_attribute <- attribute_health(
erf_shape = "log_linear",
rr_central = exdat_pm$relative_risk,
```

```
rr_increment = 10,
exp_central = exdat_pm$mean_concentration,
cutoff_central = exdat_pm$cut_off_value,
bhd_central = exdat_pm$incidence
)

results <- monetize(
  output_attribute = output_attribute,
  discount_shape = "exponential",
  discount_rate = 0.03,
  n_years = 5,
  valuation = 50000 # E.g. EURO
)

# Attributable COPD cases its monetized impact
results$monetization_main |>
  dplyr::select(impact, monetized_impact)
```

`multiexpose` *Aggregate health impacts from multiple exposures*

Description

This function aggregates health impacts from multiple exposures to environmental stressors.

Usage

```
multiexpose(
  output_attribute_exp_1,
  output_attribute_exp_2,
  exp_name_1,
  exp_name_2,
  approach_multiexposure = "additive"
)
```

Arguments

`output_attribute_exp_1`, `output_attribute_exp_2`	Output of attribute() for exposure 1 and 2, respectively. Baseline health data and population must be identical in outputs 1 and 2.
`exp_name_1`, `exp_name_2`	`String` referring to the name of the environmental exposures 1 and 2
`approach_multiexposure`	`String` specifying the multiple exposures approach to be used in the assessment. Options: "additive" (default), "multiplicative" or "combined".

Details

Methodology

This function can add up the attributable health impacts from correlated exposures applying one of the following methods (Strak et al. 2024):

- Additive (Steenland and Armstrong 2006)
- Multiplicative (Jerrett et al. 2013)
- Combined (Steenland and Armstrong 2006)

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- Two correlated exposures

Value

This function returns a `list` containing:

1) `health_main` (`tibble`) containing the main results;

- `impact` (`numeric` column) attributable health burden/impact
- `pop_fraction` (`numeric` column) population attributable fraction; only applicable in relative risk assessments
- And many more

2) `health_detailed` (`list`) containing detailed (and interim) results.

- `input_args` (`list`) containing all the argument inputs used in the background
- `input_table` (`tibble`) containing the inputs after preparation
- `results_raw` (`tibble`) containing results for all combinations of input (geo units, uncertainty, age and sex specific data...)
- `results_by_...` (`tibble`) containing results stratified by each geographic unit, age or sex.

Author(s)

Alberto Castro & Axel Luyten

References

Jerrett M, Burnett RT, Beckerman BS, Turner MC, Krewski D, Thurston G, Martin RV, van Donkelaar A, Hughes E, Shi Y, Gapstur SM, Thun MJ, Pope 3CA (2013). "Spatial analysis of air pollution and mortality in California." *American Journal of Respiratory and Critical Care Medicine*, **188**(5), 593–599. doi:10.1164/rccm.201303060 9OC.

Steenland K, Armstrong B (2006). "An overview of methods for calculating the burden of disease due to specific risk factors." *Epidemiology*, **17**(5), 512–519. doi:10.1097/01.ede.0000229155.05644.43.

Strak M, Houthuijs D, Staatsen B (2024). "D1.2 Report on the methodology for assessing the burden of correlated exposures." EU Project BEST-COST.

See Also

- Upstream: `attribute_health`, `attribute_lifetable`

Examples

```
# Goal: determine aggregated health impacts from multiple exposures
# Step 1: create assessment with exposure 1
output_attribute_exp_1 <- attribute_health(
  erf_shape = "log_linear",
  rr_central = 1.369,
  rr_increment = 10,
  exp_central = 8.85,
  cutoff_central = 5,
  bhd_central = 30747
)
output_attribute_exp_1$health_main$impact
# Step 2: create assessment with exposure 2
output_attribute_exp_2 <- attribute_mod(
  output_attribute = output_attribute_exp_1,
  exp_central = 10.9,
  rr_central = 1.031
)
output_attribute_exp_2$health_main$impact
# Step 3: aggregate impacts of the two assessments
results <- multiexpose(
  output_attribute_exp_1 = output_attribute_exp_1,
  output_attribute_exp_2 = output_attribute_exp_2,
  exp_name_1 = "pm2.5",
  exp_name_2 = "no2",
  approach_multiexposure = "multiplicative"
)
results$health_main$impact
```

`prepare_exposure` *Prepare exposure data*

Description

This function prepares tabular population exposure data compatible with the `attribute()` and `compare()` functions, based on gridded pollution concentration data and polygon data representing geographic units. If population data is provided, the function calculates an average concentration value in each geographic unit that is weighted with the population number at each location. If no population data is provided, the function calculates the simple spatial average concentration in each geographic unit.

Usage

```
prepare_exposure(
  poll_grid,
  geo_units,
  population = NULL,
  pop_grid = NULL,
  geo_id_micro = NULL,
  geo_id_macro = NULL,
  bin_width = 0.1
)
```

Arguments

`poll_grid`	`SpatRaster` of the pollution concentration data.
`geo_units`	`sf` of the geographic units or sub-units.
`population`	`Integer vector` of the total population number in each geographic sub-unit.
`pop_grid`	`SpatRaster` of the gridded population data.
`geo_id_micro`	`Numeric` or `string vector` of the IDs of the geographic units. Required if `pop_grid` is given or if no population data is provided.
`geo_id_macro`	`Numeric` or `string vector` of the higher-level IDs of the geographic units the sub-unit belong to and will be aggregated at. Required if `population` is provided.
`bin_width`	`Numeric` specifying the width of the population exposure bins.

Value

This function returns a `list` containing:

1) `main` (`list`) containing the main results as vectors;

- `geo_id_micro` of `geo_id_macro` (`string` column) containing the (higher-level) geographic IDs of the assessment
- `exposure_mean` (`numeric` column) containing the (population-weighted) mean exposure
- `population_total` (`integer` column) containing the total population in each geographic unit, if population data was provided

2) `detailed` (`list`) containing detailed (and interim) results.

Author(s)

Arno Pauwels & Liliana Vazquez Fernandez

Examples

```
# Goal: determine population-weighted mean PM2.5 exposure for several
# neighborhoods of Brussels (Belgium)

path <- system.file("extdata", "exdat_pwm_1.tif", package = "healthiar")
```

```
exdat_pwm_1 <- terra::rast(path)

pwm <- prepare_exposure(
  poll_grid = exdat_pwm_1, # Formal class SpatRaster
  geo_units = exdat_pwm_2, # sf of the geographic sub-units
 population = sf::st_drop_geometry(exdat_pwm_2$population), # population per geographic sub-unit
 geo_id_macro = sf::st_drop_geometry(exdat_pwm_2$region) # higher-level IDs to aggregate at
)

pwm$exposure_main # population-weighted mean exposures for the (higher-level) geographic units
```

`prepare_lifetable` *Convert multi-year life table to single year life table*

Description

This function determines populations and deaths by one year age groups.

Usage

```
prepare_lifetable(age_group, population, bhd)
```

Arguments

`age_group`	`Numeric vector` referring to the first years of the age groups. E.g. c(0, 20, 40, 60) means [0, 20), [20, 40), [40, 60), [60,)
`population`	`Numeric vector` referring to mid-year populations by age group.
`bhd`	`Numeric vector` referring to the baseline health data (deaths) by age group.

Details

Methodology

The conversion follows the methodology of the WHO tool. See the AirQ+ manual "Health impact assessment of air pollution: AirQ+ life table manual" for guidance on how to convert larger age groups to 1 year age groups, section "Estimation of yearly values" (WHO 2020).

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- YLL and deaths with life table

Value

This function returns a `tibble` containing the columns:

- `population_for_attribute` (`numeric`) containing population values for each age
- `bhd_for_attribute` (`numeric`) containing baseline health data values for each age
- and more columns containing input data or results

Author(s)

Alberto Castro & Axel Luyten

References

WHO (2020). “Health impact assessment of air pollution: AirQ+ life table manual.” World Health Organization - Regional Office for Europe. https://www.who.int/europe/publications/i/item/WHO-EURO-2020-1559-41310-56212.

See Also

- Downstream: `attribute_lifetable`

Examples

```
# Goal: Convert 5-year population and death data into single year life table
results <- prepare_lifetable(
  age_group = c(0, 5, 10, 15),
  population = c(3387900, 3401300, 3212300, 3026100),
  bhd = c(4727, 472, 557, 1323)
)
```

`prepare_mdi` *Create the BEST-COST Multidimensional Deprivation Index (MDI)*

Description

This function creates the BEST-COST Multidimensional Deprivation Index (MDI) and checks internal consistency of the single deprivation indicators using Cronbach’s coefficient α and other internal consistency checks

Usage

```
prepare_mdi(
  geo_id_micro,
  edu,
  unemployed,
  single_parent,
  pop_change,
  no_heating,
  n_quantile,
  verbose = TRUE
)
```

Arguments

`geo_id_micro`	`Numeric vector` or `string vector` specifying the unique ID codes of each geographic area considered in the assessment (`geo_id_micro`).
`edu`	`Numeric vector` indicating educational attainment as % of individuals (at the age 18 or older) without a high school diploma (ISCED 0-2) per geo unit
`unemployed`	`Numeric vector` containing % of unemployed individuals in the active population (18-65) per geo unit
`single_parent`	`Numeric vector` containing single-parent households as % of total households headed by a single parent per geo unit
`pop_change`	`Numeric vector` containing population change as % change in population over the previous 5 years (e.g., 2017-2021) per geo unit
`no_heating`	`Numeric vector` containing % of households without central heating per geo unit
`n_quantile`	`Integer value` specifying the number of quantiles in the analysis.
`verbose`	`Boolean` indicating whether function output is printed to console. Default: `TRUE`.

Details

Methodology

This function condenses socio-economic indicators into a multiple deprivation index (MDI) (Mogin et al. 2025). The reliability of the MDI is assessed using Cronbach's alpha (Cronbach 1951).

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- Multiple deprivation index

Data completeness and imputation

Ensure the data set is as complete as possible. Otherwise, you can try to impute missing data, but R^2 should be greater than or equal to 0.7.

Plots

See the example below for how to reproduce the box plots and the histogram after the `prepare_mdi` function call.

Value

This function returns a `list` containing

1) `mdi_main` (`tibble`) with the columns (selection);

- `geo_id_micro` containing the `numeric` geo id's
- `MDI` containing the `numeric` BEST-COST Multidimensional Deprivation Index values
- `MDI_index` `numeric` decile based on values in the column `MDI`
- additional columns containing the function input data

2) `mdi_detailed` (`list`) with several elements for the internal consistency check of the BEST-COST Multidimensional Deprivation Index.

- `boxplot` (`language`) containing the code to reproduce the boxplot of the single indicators
- `histogram` (`language`) containing the code to reproduce a histogram of the BEST-COST Multidimensional Deprivation Index (MDI) values with a normal distribution curve
- `descriptive_statistics` (`list` table of descriptive statistics (mean, SD, min, max) of the normalized input data and the MDI
- `cronbachs_alpha_value` (`numeric value` See the Details section for the reliability rating this value indicates
- `pearsons_corr_coeff` (`numeric vector`) Person's correlation coefficient (pairwise-comparisons)

Author(s)

Carl Baravelli, Vanessa Gorasso, Alberto Castro & Axel Luyten

References

Cronbach LJ (1951). "Coefficient alpha and the internal structure of tests." *Psychometrika*, **16**(3), 297–334. doi:10.1007/BF02310555.

Mogin G, Gorasso V, Idavain J, Lepnurm M, Delaunay-Havard S, Kocbach Bølling A, Buekers J, Luyten A, Devleesschauwer B, Baravelli CM (2025). "A scoping review of multiple deprivation indices in Europe." *European Journal of Public Health*, **35**(6), 1122-1128. doi:10.1093/eurpub/ckaf190, https://academic.oup.com/eurpub/article-pdf/35/6/1122/65042936/ckaf190.pdf.

See Also

- Downstream: `socialize`

Examples

```
# Goal: create the BEST-COST Multidimensional Deprivation Index for
# a selection of geographic units

results <- prepare_mdi(
  geo_id_micro = exdat_prepare_mdi$id,
  edu = exdat_prepare_mdi$edu,
  unemployed = exdat_prepare_mdi$unemployed,
  single_parent = exdat_prepare_mdi$single_parent,
  pop_change = exdat_prepare_mdi$pop_change,
  no_heating = exdat_prepare_mdi$no_heating,
  n_quantile = 10,
  verbose = TRUE
)

results$mdi_main |>
  dplyr::select(geo_id_micro, MDI, MDI_index) |>
  dplyr::slice(1:15)
```

```
# Reproduce plots after the function call
eval(results$mdi_detailed$boxplot)
eval(results$mdi_detailed$histogram)
```

`socialize` *Consider socio-economic aspects in the attributable health impacts*

Description

This function analyzes differences in attributable health impacts across study areas looking at the value of a socio-economic indicator (e.g. multiple deprivation index). If nothing is entered in the argument `output_attribute`, it is assumed that all data come from a table and the argument refer to the columns of that table.

Usage

```
socialize(
  output_attribute = NULL,
  age_group,
  geo_id_micro,
  social_indicator = NULL,
  increasing_deprivation = TRUE,
  n_quantile = NULL,
  social_quantile = NULL,
  population = NULL,
  ref_prop_pop = NULL,
  impact = NULL,
  exp = NULL,
  bhd = NULL,
  pop_fraction = NULL
)
```

Arguments

`output_attribute`	`List` containing the outputs of the `healthiar::attribute_health()` assessments for each age group (each list element should be an age group-specific assessment).
`age_group`	`String vector` with the age groups included in the age standardization. The vector refers to age-dependent data in this function and to `output_attribute` (if provided).
`geo_id_micro`	`Numeric vector` or `string vector` specifying the unique ID codes of each geographic area considered in the assessment (`geo_id_micro`).

`social_indicator`	`Numeric vector` showing the social indicator used for the analysis, e.g. a deprivation score (indicator of economic wealth) for each geographic unit. The length and the values must correspond with `geo_id_micro`. If `geo_id_micro` is not entered when using argument `output_attribute`, `social_indicator` must correspond to the column `geo_id_micro` in `results_by_age_group` of `output_attribute`.
`increasing_deprivation`	`Boolean` variable (`TRUE`/`FALSE`) specifying whether an increase in `social_indicator` corresponds to an increase (`TRUE`) or decrease `FALSE` in deprivation. Default: `TRUE`.
`n_quantile`	`Integer value` specifying the number of quantiles in the analysis.
`social_quantile`	`Integer vector` showing the values from 1 to the number of quantiles assigned to each geographic unit. Either enter `social_indicator` and `n_quantile` or `social_quantile`
`population`	`Numeric vector` specifying the population by age group and geographic unit.
`ref_prop_pop`	`Numeric vector` specifying with the reference proportion of population for each age group. If this argument is empty, the proportion of `population` by age group in the provided data will be used.
`impact`	*(only if* `output_attribute` *not specified)* `Numeric vector` containing the attributable health impacts by both age group and geo id.
`exp`	*(only if* `output_attribute` *not specified)* `Numeric vector` specifying the exposure level(s) to the environmental stressor.
`bhd`	*(only if* `output_attribute` *not specified)* `Numeric vector` specifying the baseline health data of the health outcome of interest per age group.
`pop_fraction`	*(only if* `output_attribute` *not specified)* `Numeric vector` specifying the population attributable fraction by age group and geographic unit.

Details

Methodology

This function estimates the absolute and relative differences in attributable health impacts comparing study areas with different values for a socio-economic indicator (Renard et al. 2019).

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- Health impact attributable to social indicator

Value

This function returns a `list` containing the impact (absolute and relative) theoretically attributable to the difference in the social indicator (e.g. degree of deprivation) between the quantiles:

1) `social_main` (`tibble`) containing the main results;

- `difference_value` (`numeric` column) attributable health burden/impact due to differences in deprivation levels

- And more

2) `social_detailed` (`list`) containing detailed (and interim) results.

- `input_data_with_quantile` (`tibble`) containing input data and information about the social quantile
- `results_all_parameters` (`tibble`) containing deprivation-related results
- `parameters_overall` (`tibble`) containing overall results for different input variables
- `parameters_per_quantile` (`tibble`) containing quantile-specific results for different input variables

If the argument `output_attribute` was specified, then the two lists are added next to the existing attribute output.

Author(s)

Alberto Castro & Axel Luyten

References

Renard F, Devleesschauwer B, Speybroeck N, Deboosere P (2019). "Monitoring health inequalities when the socio-economic composition changes: are the slope and relative indices of inequality appropriate? Results of a simulation study." *BMC Public Health*, **19**(1), 662. ISSN 1471-2458, doi:10.1186/s1288901969801.

See Also

- Upstream: `attribute_health`, `attribute_lifetable`, `prepare_mdi`,

Examples

```
# Goal: determine fraction of attributable health impact that can
# be attributed to differences in deprivation between the geographic
# units under analysis

## Create assessments for multiple geographic units for the age group
## 40 years and younger
results_age_groups <-
   healthiar::attribute_health(
     age_group = exdat_socialize$age_group,
     exp_central = exdat_socialize$pm25_mean,
     cutoff_central = 0,
     rr_central = exdat_socialize$rr,
     erf_shape = "log_linear",
     rr_increment = 10,
     bhd_central = exdat_socialize$mortality,
     population = exdat_socialize$population,
     geo_id_micro = exdat_socialize$geo_unit)

## Difference in attributable impacts between geographic units
## that is attributable to differences in deprivation
```

```
results <- socialize(
  output_attribute = results_age_groups,
  age_group = exdat_socialize$age_group, # The same as in attribute_health()
  ref_prop_pop = exdat_socialize$ref_prop_pop,
  geo_id_micro = exdat_socialize$geo_unit,
  social_indicator = exdat_socialize$score,
  n_quantile = 10,
  increasing_deprivation = TRUE)


results$social_main |>
  dplyr::filter(difference_type == "relative") |>
  dplyr::filter(difference_compared_with == "overall") |>
  dplyr::select(first, last, difference_type, difference_value, comment)
```

`standardize` *Obtain age-standardized health impacts*

Description

This function obtains age-standardized health impacts based on multiple age-group specific assessments

Usage

```
standardize(output_attribute, age_group, ref_prop_pop = NULL)
```

Arguments

`output_attribute`	`List` containing the outputs of the `healthiar::attribute_health()` assessments for each age group (each list element should be an age group-specific assessment).
`age_group`	`String vector` with the age groups included in the age standardization. The vector refers to age-dependent data in this function and to `output_attribute` (if provided).
`ref_prop_pop`	`Numeric vector` specifying with the reference proportion of population for each age group. If this argument is empty, the proportion of `population` by age group in the provided data will be used.

Details

Methodology

This function applies the direct method of standardization, where the age-specific rates observed in a study population are applied to a standard (reference) population distribution.

For age standardization in health impact assessments, the World Health Organization (Ahmad et al. 2001) and the Global Burden of Disease study (GBD 2019 Demographics Collaborators 2020) provide the relevant information on this topic.

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- Standardization

This function works after running `attribute_health()` or `attribute_lifetable()` functions. If you want to use it in combination with compare(), please standardize first the results of attribute functions and then compare.

Value

This function returns a `list` containing:

1) `health_main` (`tibble`) containing the age-standardized main results;

2) `health_detailed` (`tibble`) containing the results per age group.

Author(s)

Alberto Castro & Axel Luyten

References

Ahmad OB, Pinto CB, Lopez AD, Murray CJ, Lozano R, Inoue M (2001). "Age standardization of rates: a new WHO standard." Technical Report GPE Discussion Paper Series: No. 31, World Health Organization, Geneva.

GBD 2019 Demographics Collaborators (2020). "Global age-sex-specific fertility, mortality, healthy life expectancy (HALE), and population estimates in 204 countries and territories, 1950-2019: a comprehensive demographic analysis for the Global Burden of Disease Study 2019." *The Lancet*, **396**(10258), 1160-1203. doi:10.1016/S01406736(20)309776.

See Also

- Upstream: `attribute_health`, `attribute_lifetable`

Examples

```
# Goal: age-standardize two age group-specific impacts
output_attribute <- attribute_health(
  rr_central = 1.063,
  rr_increment = 10,
  erf_shape = "log_linear",
  cutoff_central =  0,
  age_group = c("below_40", "above_40"),
  exp_central = c(8.1, 10.9),
  bhd_central = c(1000, 4000),
  population = c(100000, 500000)
)
```

```
results <- standardize(
  output_attribute = output_attribute,
  age_group = c("below_40", "above_40"),
  ref_prop_pop = c(0.5, 0.5)
)
results$health_detailed$results_raw$impact_per_100k_inhab # age group-specific impact rate
results$health_main$impact_per_100k_inhab # age-standardized impact rate
```

`summarize_uncertainty` *Get Monte Carlo confidence intervals*

Description

This function obtains a summary of uncertainty (based on central, lower and upper estimates of at least one input variable) using a Monte Carlo simulation.

Input variables that will be processed are:

- relative_risk (`rr_...`)
- exposure (`exp_...`)
- cutoff (`cutoff_...`)
- baseline health data (`bhd_...`)
- disability weight (`dw_...`)
- duration (`duration_...`)

Usage

```
summarize_uncertainty(output_attribute, n_sim, seed = NULL)
```

Arguments

`output_attribute`	`variable` in which the output of a `healthiar::attribute_...()` function call are stored.
`n_sim`	`numeric value` indicating the number of simulations to be performed.
`seed`	`numeric value` for fixing the randomization. Based on it, each geographic unit is assigned a different. If empty, 123 is used as the base seed per default. The function preserves and restores the user's original random seed (if set prior to calling the function) upon function completion.

Details

Function arguments `seed` If the `seed` argument is specified then the `parallel` package is used to generate independent L'Ecuyer random number streams. One stream is allocated per variable (or per variable–geography combination, as needed), ensuring reproducible and independent random draws across variables and scenarios.

Methodology

This function summarizes the uncertainty of the attributable health impacts (i.e. a single confidence interval instead of many combinations). For this purpose, it employs a Monte Carlo simulation methodology (Robert and Casella 2004) and framework application (Rubinstein and Kroese 2016).

Detailed information about the methodology (including equations) is available in the package vignette. More specifically, see chapters:

- Monte Carlo simulation

Value

This function returns a `list` containing:

1) `uncertainty_main` (`tibble`) containing the `numeric` summary uncertainty central estimate and corresponding lower and upper confidence intervals for the attributable health impacts obtained through Monte Carlo simulation;

2) `uncertainty_detailed` (`list`) containing detailed (and interim) results.

- `impact_by_sim` (`tibble`) containing the results for each simulation
- `uncertainty_by_geo_id_micro` (`tibble`) containing results for each geographic unit under analysis (specified in `geo_id_micro` argument in the preceding `attribute_health` call)

The two results elements are added to the existing output.

Author(s)

Alberto Castro & Axel Luyten

References

Robert CP, Casella G (2004). *Monte Carlo Statistical Methods*, Springer Texts in Statistics. Springer Science and Business Media. doi:10.1007/9781475741452.

Rubinstein RY, Kroese DP (2016). *Simulation and the Monte Carlo Method*. John Wiley and Sons. doi:10.1002/9781118631980.

See Also

- Upstream: `attribute_health`, `attribute_lifetable`, `compare`

Examples

```
# Goal: obtain summary uncertainty for an existing attribute_health() output
# First create an assessment
attribute_health_output <- attribute_health(
  erf_shape = "log_linear",
  rr_central = 1.369,
  rr_lower = 1.124,
  rr_upper = 1.664,
  rr_increment = 10,
  exp_central = 8.85,
  exp_lower = 8,
  exp_upper = 10,
  cutoff_central = 5,
  bhd_central = 30747,
  bhd_lower = 28000,
  bhd_upper = 32000
)
# Then run Monte Carlo simulation
results <- summarize_uncertainty(
  output_attribute = attribute_health_output,
  n_sim = 100
)
results$uncertainty_main$impact # Central, lower and upper estimates
```

Index