Package ‘healthiar’

November 11, 2025

Title Quantify and Monetize the Burden of Disease Attributable to Exposure

Version 0.2.1

Description This R package has been developed with a focus on air pollution and noise but can 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.
References (in chronological order):
WHO (2003a) ``Assessing the environmental burden of disease at national and local levels'' <https://www.who.int/publications/i/item/9241546204> (accessed October 2025);
WHO (2003b) ``Comparative quantification of health risks: Conceptual framework and methodological issues'' <doi:10.1186/1478-7954-1-1> (accessed October 2025);
Miller & Hurley (2003) ``Life table methods for quantitative impact assessments in chronic mortality'' <doi:10.1136/jech.57.3.200> (accessed October 2025);
Steenland & Armstrong (2006) ``An Overview of Methods for Calculating the Burden of Disease Due to Specific Risk Factors'' <doi:10.1097/01.ede.0000229155.05644.43> (accessed October 2025);
Miller (2010) ``Report on estimation of mortality impacts of particulate air pollution in London'' <https://cleanair.london/app/uploads/CAL-098-Mayors-health-study-report-June-2010-1.pdf> (accessed October 2025);
WHO (2011) ``Burden of disease from environmental noise'' <https://iris.who.int/items/723ab97c-5c33-4e3b-8df1-744aa5bc1c27> (accessed October 2025);
Jerrett et al. (2013) ``Spatial Analysis of Air Pollution and Mortality in California'' <doi:10.1164/rccm.201303-0609OC> (accessed October 2025);
GBD 2019 Risk Factors Collaborators (2020) ``Global burden of 87 risk factors in 204 countries and territories, 1990–2019'' <doi:10.1016/S0140-6736(20)30752-2> (accessed October 2025);
VanderWeele (2019) ``Optimal Approximate Conversions of Odds Ratios and Hazard Ratios to Risk Ratios'' <doi:10.1111/biom.13197> (accessed October 2025);
WHO (2020) ``Health impact assessment of air pollution: AirQ+ life table manual'' <https://iris.who.int/bitstream/handle/10665/337683/WHO-EURO-2020-1559-41310-56212-eng.pdf?sequence=1> (accessed October 2025);
ETC HE (2022) ``Health risk assessment of air pollution and the impact of the new WHO guidelines'' <https://www.eionet.europa.eu/etcs/all-etc-reports> (accessed October 2025);
Kim et al. (2022) ``DALY Estimation Approaches: Understanding and Using the Incidence-based Approach and the Prevalence-

based Approach" <doi:10.3961/jpmph.21.597> (accessed October 2025);
Pozzer et al. (2022) ``Mortality Attributable to Ambient Air Pollution: A Review of Global Estimates" <doi:10.1029/2022GH000711> (accessed October 2025);
Teaching group in EBM (2022) ``Evidence-based medicine research helper" <https://ebm-helper.cn/en/Conv/HR_RR.html> (accessed October 2025).

License GPL (>= 3)

Encoding UTF-8

RoxygenNote 7.3.3

Imports dplyr, purrr, tibble, tidyr

Depends R (>= 4.2.0)

LazyData true

Config/testthat/edition 3

VignetteBuilder knitr

Suggests sf, terra, testthat, devtools, knitr, 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>),
Vanessa Gorasso [ctb] (ORCID: <https://orcid.org/0000-0001-6884-9316>),
Carl Michael Baravelli [ctb] (ORCID: <https://orcid.org/0000-0001-7772-5315>),
Susanne Breitner [ctb] (ORCID: <https://orcid.org/0000-0002-0956-6911>),
Maria Lepnurm [ctb] (ORCID: <https://orcid.org/0009-0009-4372-6227>),
Maria Jose Rueda Lopez [ctb] (ORCID: <https://orcid.org/0000-0002-2443-1038>),
Iracy Pimenta [ctb] (ORCID: <https://orcid.org/0000-0003-0032-1536>),
Andreia Novais [ctb] (ORCID: <https://orcid.org/0009-0007-7775-108X>),
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 2025-11-11 10:00:09 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). If the value of `duration_central` is 1 year, it refers to the prevalence-based approach, while a value above 1 year to the incidence-based approach (Kim et al. 2022, https://doi.org/10.3961/jpmph.21.597).

Assessment of multiple geographic units

To assess the attributable health impact/burden across multiple geographic units with `attribute_health()`, you must specify the argument `geo_id_micro` and (optionally) `geo_id_macro`, in addition to the other required function arguments.

The length of the input vectors to the function arguments must be:

$$\text{length input vectors} = \text{number of geo units} \times \text{number of exposure categories}$$

$$(\times \text{number of age groups (if entered)} \times \text{number of sex groups (if entered)})$$

I.e. there must be one line / observation for each specific combination of geo unit, exposure category, age and sex group.

Alternatively, for those arguments that are independent of location (e.g. `approach_risk`, `rr_...`, `erf_shape`, ...), you can enter a single value, which will be recycled to match the length of the other geo unit-specific input data. Additional categories can be passed on via the `info` argument.

Equations (relative risk)

The most general equation describing the population attributable fraction (PAF) mathematically is an integral form (GBD 2019 Risk Factors Collaborators 2020, https://doi.org/10.1016/S0140-6736(20)30752-2):

$$PAF = \frac{\int RR(x)PE(x)dx - 1}{\int RR(x)PE(x)dx}$$

Where:

x = exposure level

$PE(x)$ = population distribution of exposure

$RR(x)$ = relative risk at exposure level compared to the reference level

If the population exposure is described as a categorical rather than continuous exposure, the integrals in this equation may be converted to sums, resulting in the following equation for the PAF (WHO 2003a, https://www.who.int/publications/i/item/9241546204; WHO 2011, https://iris.who.int/items/723ab97c-5c33-4e3b-8df1-744aa5bc1c27):

$$PAF = \frac{\sum RR_i \times PE_i - 1}{\sum RR_i \times PE_i}$$

Where:

i = is the exposure category (e.g. in bins of 1 $\mu g/m^3$ PM2.5 or 5 dB noise exposure)

PE_i = fraction of population in exposure category i

RR_i = relative risk associated with the mean exposure level in exposure category i compared to the reference level

There is one alternative for the PAF for categorical exposure distribution that is commonly used, which is mathematically equivalent to the equation right above, meaning that numerical estimates based on these equations are identical (WHO 2003b, https://doi.org/10.1186/1478-7954-1-1; WHO 2011, https://iris.who.int/items/723ab97c-5c33-4e3b-8df1-744aa5bc1c27):

$$PAF = \frac{\sum PE_i(RR_i - 1)}{\sum PE_i(RR_i - 1) + 1}$$

Where:

i = is the exposure category (e.g. in bins of 1 $\mu g/m^3$ PM2.5 or 5 dB noise exposure)

PE_i = fraction of population in exposure category i

RR_i = relative risk associated with the mean exposure level in exposure category i compared to the reference level

Finally, if the exposure is provided as the population weighted mean concentration (PWC), the equation for the PAF is reduced to (ETC HE 2022, https://www.eionet.europa.eu/etcs/all-etc-reports:

$$PAF = \frac{RR_{PWC} - 1}{RR_{PWC}}$$

Where RR_{PWC} is the relative risk associated with the population weighted mean exposure.

Equation (absolute risk)

$$N = \sum AR_i \times PE_i$$

Where:

N = the number of cases of the exposure-specific health outcome that are attributed to the exposure

AR_i = absolute risk associated with the mean exposure level of exposure category i

PE_i = population exposed (absolute number) to exposure levels of exposure category i

Conversion of alternative risk measures to relative risks For conversion of hazard ratios and/or odds ratios to relative risks refer to VanderWeele 2019 (https://doi.org/10.1111/biom.13197) and/or use the conversion tools developed by the Teaching group in EBM in 2022 for hazard ratios (https://ebm-helper.cn/en/Conv/HR_RR.html) and/or odds ratios (https://ebm-helper.cn/en/Conv/OR_RR.html).

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.

- `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_table` (`tibble`) containing the inputs to each relevant argument
- `input_args` (`list`) containing all the argument inputs used in the background

Author(s)

Alberto Castro & Axel Luyten

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

# Goal: determine mean attributable health impacts by education level
info <- data.frame(
  education = rep(c("secondary", "bachelor", "master"), each = 5) # education level
)
output_attribute <- attribute_health(
  rr_central = 1.063,
  rr_increment = 10,
  erf_shape = "log_linear",
  cutoff_central =  0,
  exp_central = sample(6:10, 15, replace = TRUE),
  bhd_central = sample(100:500, 15, replace = TRUE),
  geo_id_micro = c(1:nrow(info)), # a vector of (random) unique IDs must be entered
  info = info
)
output_stratified <- output_attribute$health_detailed$results_raw |>
 dplyr::group_by(info_column_1) |>
 dplyr::summarize(mean_impact = mean(impact)) |>
 print()
```

`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.

`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.

Conversion of multi-year to single year age groups

To convert multi-year/larger age groups to 1 year age groups use the function `prepare_lifetable()` and see its function documentation for more info.

Life table methodology

The life table methodology of `attribute_lifetable()` follows that of the WHO tool AirQ+, and is described in more detail by Miller & Hurley (2003, https://doi.org/10.1136/jech.57.3.200).

In short, two scenarios are compared: 1) a scenario with the exposure level specified in the function ("exposed scenario") and 2) a scenario with no exposure ("unexposed scenario"). First, the entry and mid-year populations of the (first) year of analysis in the unexposed scenario is determined using modified survival probabilities. Second, age-specific population projections using scenario-specific survival probabilities are done for both scenarios. Third, by subtracting the populations in the unexposed scenario from the populations in the exposed scenario the premature deaths/years of life lost attributable to the exposure are determined.

An expansive life table case study by Miller (2010) is available here: https://cleanair.london/app/uploads/CAL-098-Mayors-health-study-report-June-2010-1.pdf.

Determination of populations in the (first) year of analysis

The entry (i.e. start of year) populations in both scenarios is determined as follows:

$$entry\_population_{year_1} = midyear\_population_{year_1} + \frac{deaths_{year_1}}{2}$$

Exposed scenario The survival probabilities in the exposed scenario from start of year i to start of year i+1 are calculated as follows:

$$prob\_survival = \frac{midyear\_population_i - \frac{deaths_i}{2}}{midyear\_population_i + \frac{deaths_i}{2}}$$

Analogously, the probability of survival from start of year i to mid-year i:

$$prob\_survival\_until\_midyear = 1 - \frac{1 - prob\_survival}{2}$$

Unexposed scenario The survival probabilities in the unexposed scenario are calculated as follows:

First, the age-group specific hazard rate in the exposed scenario is calculated using the inputted age-specific mid-year populations and deaths.

$$hazard\_rate = \frac{deaths}{mid\_year\_population}$$

Second, the hazard rate is multiplied with the modification factor ($= 1 - PAF$) to obtain the age-specific hazard rate in the unexposed scenario.

$$hazard\_rate\_mod = hazard\_rate \times modification\_factor$$

Third, the the age-specific survival probabilities (from the start until the end in a given age group) in the unexposed scenario are calculated as follows (cf. Miller & Hurley 2003):

$$prob\_survival\_mod = \frac{2 - hazard\_rate\_mod}{2 + hazard\_rate\_mod}$$

Then the mid-year populations of the (first) year of analysis (year_1) in the unexposed scenario are determined as follows:

First, the survival probabilities from start of year i to mid-year i in the unexposed scenario is calculated as:

$$prob\_survival\_until\_midyear\_mod = 1 - \frac{1 - prob\_survival\_mod}{2}$$

Second, the mid-year populations of the (first) year of analysis (year_1) in the unexposed scenario is calculated:

$$midyear\_population\_unexposed_{year_1} = entry\_population_{year_1} \times prob\_survival\_until\_midyear_{mod}$$

Population projection

Using the age group-specific and scenario-specific survival probabilities calculated above, future populations of each age-group under each scenario are calculated.

Unexposed scenario The entry and mid-year population projections of in the exposed scenario is done as follows:

First, the entry population of year i+1 is calculated (which is the same as the end of year population of year i) by multiplying the entry population of year i and the modified survival probabilities.

$$entry\_population_{i+1} = entry\_population_i \times prob\_survival\_mod$$

Second, the mid-year population of year i+1 is calculated.

$$midyear\_population_{i+1} = entry\_population_{i+1} \times prob\_survival\_until\_midyear$$

Exposed scenario The population projections for the two possible options of `approach_exposure` (`"single_year"` and `"constant"`) for the unexposed scenario are different. In the case of `"single_year"` exposure, the population projection for the years after the year of exposure is the same as in the unexposed scenario.

In the case of `"constant"` the population projection is done as follows:

First, the entry population of year i+1 is calculated (which is the same as the end of year population of year i) using the entry population of year i.

$$entry\_population_{i+1} = entry\_population_i \times prob\_survival$$

Second, the mid-year population of year i+1 is calculated.

$$midyear\_population_{i+1} = entry\_population_{i+1} \times prob\_survival\_until\_midyear$$

Conversion of alternative risk measures to relative risks

For conversion of hazard ratios and/or odds ratios to relative risks refer to https://doi.org/10.1111/biom.13197 and/or use the conversion tool for hazard ratios (https://ebm-helper.cn/en/Conv/HR_RR.html) and/or odds ratios (https://ebm-helper.cn/en/Conv/OR_RR.html).

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.

- `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)
- `results_by_year` (`tibble`) containing results by year
- `results_by_sex` (`tibble`) containing results by sex
- `results_by_age_group` (`tibble`) containing results by age group
- `intermediate_calculations` (`tibble`) containing intermediate results, among others population projections (for both the exposed and unexposed scenarios) and impact by age and year stored in nested `tibbles`
- `input_table` (`tibble`) containing the inputs to each relevant argument
- `input_args` (`list`) containing all the argument inputs used in the background

Author(s)

Alberto Castro & Axel Luyten

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 assesses the attributable health impacts in a new scenario 2 which is obtained by modifying an existing scenario 1. Supply an existing attribute output and specify how scenario 1 should be modified to create 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

Please see the function documentation of `attribute_health` for the methods used.

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.

- `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_table` (`tibble`) containing the inputs to each relevant argument
- `input_args` (`list`) containing all the argument inputs used in the background

Author(s)

Alberto Castro & Axel Luyten

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

Usage

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

Arguments

`output_attribute`	`List` produced by `healthiar::attribute()` 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). Only to be entered if nominal (not real) discount rate is entered in the function. Default value = NULL (assuming no nominal discount rate).
`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

Equation cost-benefit analysis

$$net\_benefit = benefit - cost$$

$$cost\_benefit\_ratio = \frac{benefit}{cost}$$

$$return\_on\_investment = \frac{benefit - cost}{cost} \times 100$$

For the equations regarding the monetization of the cost and the benefit please see the function documentation of `monetize()`.

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

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

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).

Equations population impact fraction (PIF)

The Population Impact Fraction (PIF) is defined as the proportional change in disease or mortality when exposure to a risk factor is changed (for instance due to an intervention). The most general equation describing this mathematically is an integral form (WHO 2003a, https://www.who.int/publications/i/item/92415462 WHO 2003b, https://doi.org/10.1186/1478-7954-1-1):

$$PIF = \frac{\int RR(x)PE(x)dx - \int RR(x)PE'(x)dx}{\int RR(x)PE(x)dx}$$

Where:

x = exposure level

PE(x) = population distribution of exposure

PE'(x) = alternative population distribution of exposure

RR(x) = relative risk at exposure level compared to the reference level

If the population exposure is described as a categorical rather than continuous exposure, the integrals in equation (5) may be converted to sums, resulting in the following equations for the PIF (WHO

2003a, https://www.who.int/publications/i/item/9241546204; WHO 2003b, https://doi.org/10.1186/1478-7954-1-1):

$$PIF = \frac{\sum RR_i \times PE_i - \sum RR_i PE'_i}{\sum RR_i PE_i}$$

Where:

i = is the exposure category (e.g. in bins of 1 $\mu g/m^3$ PM2.5 or 5 dB noise exposure)

PE_i = fraction of population in exposure category i

PE'_i = fraction of population in category i for alternative (ideal) exposure scenario

RR_i = relative risk for exposure category level i compared to the reference level

Finally, if the exposure is provided as the population weighted mean concentration (PWC), the equation for the PIF is reduced to:

$$PIF = \frac{RR_{PWC} - RR_{altPWC}}{RR_{PWC}}$$

Where:

RR_{PWC} = relative risk associated with the population weighted mean exposure

RR_{PWC} = relative risk associated with the population weighted mean for the alternative exposure scenario

Delta comparison approach

With the delta comparison the difference between two scenarios is obtained by subtraction. The delta approach is suited for all comparison cases, and specifically for comparison of a situation now with a situation in the future.

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

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 population 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 calculates the disability-adjusted life years (DALY) attributable to the exposure to an environmental stressor by adding the two DALY components YLL and 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.

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

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 = 1,
  discount_shape = NULL,
  inflation_rate = NULL
)
```

Arguments

`output_attribute`	`List` produced by `healthiar::attribute()` 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. If it is a nominal discount rate, no inflation is to be entered. If it is a real discount rate, the result can be adjusted by entering inflation in this function.
`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".
`inflation_rate`	`Numeric value` between 0 and 1 referring to the annual inflation (increase of prices). Only to be entered if nominal (not real) discount rate is entered in the function. Default value = NULL (assuming no nominal discount rate).

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

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. If the argument `inflation_rate` is NULL (default), it is assumed that the discount rate is already corrected for inflation). Otherwise (if a value for `inflation_rate` is entered), the resulted discount factor is adjusted for inflation.

Usage

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

Arguments

`discount_rate`	`Numeric value` showing the discount rate for future years. If it is a nominal discount rate, no inflation is to be entered. If it is a real discount rate, the result can be adjusted by entering inflation in this function.

`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".
`inflation_rate`	`Numeric value` between 0 and 1 referring to the annual inflation (increase of prices). Only to be entered if nominal (not real) discount rate is entered in the function. Default value = NULL (assuming no nominal discount rate).

Details

Equations discount factors (without inflation)

Exponential discounting (no inflation)

$$discount\_factor = \frac{1}{(1 + discount\_rate)^{n\_years}}$$

Hyperbolic discounting Harvey (no inflation)

$$discount\_factor = \frac{1}{(1 + n\_years)^{discount\_rate}}$$

Hyperbolic discounting Mazure (no inflation)

$$discount\_factor = \frac{1}{(1 + (discount\_rate \times n\_years)}$$

Equations discount factors with inflation

Exponential discounting (with inflation)

$$discount\_and\_inflation\_factor = \frac{1}{((1 + discount\_rate) \times (1 + inflation\_rate))^{n\_years}}$$

Hyperbolic discounting Harvey (with inflation)

$$discount\_and\_inflation\_factor = \frac{1}{(1 + n\_years)^{discount\_rate} \times (1 + inflation\_rate)^{n\_years}}$$

Hyperbolic discounting Mazure (with inflation)

$$discount\_and\_inflation\_factor = \frac{1}{(1 + (discount\_rate \times n\_years) \times (1 + inflation\_rate)^{n\_years}}$$

Value

This function returns the `numeric` discount factor.

Author(s)

Alberto Castro & Axel Luyten

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)
```

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). Only to be entered if nominal (not real) discount rate is entered in the function. Default value = NULL (assuming no nominal discount rate).

Details

Equation inflation factor (without discounting)

$$inflation\_factor = (1 + inflation\_rate)^{n\_years}$$

Value

This function returns the `numeric` inflation factor.

Author(s)

Alberto Castro & Axel Luyten

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

For more information about the equations used by `get_paf` please see the function documentation of `attribute_health`.

Value

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

Author(s)

Alberto Castro & Axel Luyten

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 population impact fraction*

Description

This function calculates the population 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

For more information about the equations used by `get_pif` please see the function documentation of `compare`.

Value

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

Author(s)

Alberto Castro & Axel Luyten

Examples

```
# Goal: calculate the population 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 vector` containing the relative risk. The data frame must contain the central estimate as well as the lower and upper bound of the exposure-response function.
`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`	Equation of the user-defined exposure-response function that puts the relative risk (y) in relation with exposure (x). If the function is provided as `string`, it can only contains one variable: x (exposure). E.g. "3+x+x^2". If the function is provided as a `function`, the object should have a function class. 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")`
`cutoff`	`Numeric value` showing the cut-off exposure level in ug/m3 (i.e. the exposure level below which no health effects occur).
`exp`	Population exposure to the stressor (e.g. annual population-weighted mean).

Details

Equations for scaling of relative risk

linear ERF

$$rr\_at\_exp = 1 + \frac{(rr - 1)}{rr\_increment} \cdot (exp - cutoff)$$

log-linear ERF

$$rr\_at\_exp = e^{\frac{\log(\mathrm{rr})}{\mathrm{rr\_increment}} \cdot (\mathrm{exp} - \mathrm{cutoff})}$$

log-log ERF

$$rr\_at\_exp = (\frac{exp + 1}{cutoff + 1})^{\frac{\log(\mathrm{rr})}{\log(\mathrm{rr\_increment} + \mathrm{cutoff} + 1) - \log(cutoff + 1)}}$$

linear-log ERF

$$rr\_at\_exp = 1 + \frac{\log(\mathrm{rr} - 1)}{\log(\mathrm{rr\_increment} + \mathrm{cutoff} + 1) - \log(cutoff + 1)} \cdot \frac{\log(exp + 1)}{\log(cutoff + 1)}$$

Sources

For the log-linear, log-log and linear-log exposure-response function equations see Pozzer et al. 2022 (https://doi.org/10.1029/2022GH000711).

Value

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

Author(s)

Alberto Castro & Axel Luyten

Examples

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

`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 = 0,
  inflation_rate = NULL,
  info = NULL
)
```

Arguments

`output_attribute`	`List` produced by `healthiar::attribute()` 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. If it is a nominal discount rate, no inflation is to be entered. If it is a real discount rate, the result can be adjusted by entering inflation in this function.
`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). Only to be entered if nominal (not real) discount rate is entered in the function. Default value = NULL (assuming no nominal discount rate).

`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

Equation inflation factor (without discounting)

$$inflation\_factor = (1 + inflation\_rate)^{n\_years}$$

Equations discount factors (without inflation)

Exponential discounting (no inflation)

$$discount\_factor = \frac{1}{(1 + discount\_rate)^{n\_years}}$$

Hyperbolic discounting Harvey (no inflation)

$$discount\_factor = \frac{1}{(1 + n\_years)^{discount\_rate}}$$

Hyperbolic discounting Mazure (no inflation)

$$discount\_factor = \frac{1}{(1 + (discount\_rate \times n\_years)}$$

Equations discount factors with inflation

Exponential discounting (with inflation)

$$discount\_and\_inflation\_factor = \frac{1}{((1 + discount\_rate) \times (1 + inflation\_rate))^{n\_years}}$$

Hyperbolic discounting Harvey (with inflation)

$$discount\_and\_inflation\_factor = \frac{1}{(1 + n\_years)^{discount\_rate} \times (1 + inflation\_rate)^{n\_years}}$$

Hyperbolic discounting Mazure (with inflation)

$$discount\_and\_inflation\_factor = \frac{1}{(1 + (discount\_rate \times n\_years) \times (1 + inflation\_rate)^{n\_years}}$$

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

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

Sources

For more information on the additive and combined approaches see Steenland & Armstrong 2006 (https://doi.org/10.1097/01.ede.0000229155.05644.43).

For more information on the multiplicative approach see Jerrett et al. 2013 (https://doi.org/10.1164/rccm.201303-0609OC).

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.

- `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_table` (`tibble`) containing the inputs to each relevant argument
- `input_args` (`list`) containing all the argument inputs used in the background

Author(s)

Alberto Castro & Axel Luyten

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 vector data representing geographic units. The function calculates an average concentration value in each geographic unit, weighted by the fraction of the population in each sub-unit.

Usage

```
prepare_exposure(poll_grid, geo_units, population, geo_id_macro)
```

Arguments

`poll_grid`	`SpatRaster` of the pollution concentration data.
`geo_units`	`sf` of the geographic sub-units.
`population`	`Numeric vector` containing the total population number in each geographic sub-unit.
`geo_id_macro`	`Numeric` or `string vector` containing the higher-level IDs of the geographic units the sub-unit belong to and will be aggregated at.

Value

This function returns a `list` containing:

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

- `geo_id_macro` (`string` column) containing the (higher-level) geographic IDs of the assessment
- `exp_value` (`numeric` column) containing the (population-weighted) mean exposure
- `exp_type` (`string` column) specifying the exposure type

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

Author(s)

Arno Pauwels

Examples

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

exdat_pwm_1 <- terra::rast(system.file("extdata", "exdat_pwm_1.tif", package = "healthiar"))
exdat_pwm_2 <- sf::st_read(
    system.file("extdata", "exdat_pwm_2.gpkg", package = "healthiar"),
    quiet = TRUE
)

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$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

The conversion follows the methodology of the WHO tool which is outlined in WHO 2020 (https://iris.who.int/bitstream/hand
EURO-2020-1559-41310-56212-eng.pdf?sequence=1).

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"): https://iris.who.int/bitstream/handle/10665/337683/WHO-EURO-2020-1559-41310-56212-eng.pdf (accessed April 2025)

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

Examples

```
# Goal: Convert 5-year population and death data into single year lifetable
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`) Argument must be entered for iterations. See Details for more info.
`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

The function outputs Cronbach's α.

$\alpha \geq$ **0.9** Excellent reliability

0.8 $\leq \alpha <$ **0.9** Good reliability

0.7 $\leq \alpha <$ **0.8** Acceptable reliability

0.6 $\leq \alpha <$ **0.7** Questionable reliability

α **< 0.6** Poor reliability

Data completeness and imputation: ensure the dataset is as complete as possible. You can try to impute missing data:

- Time-Based Imputation: Use linear regression based on historical trends if prior years' data is complete.
- Indicator-Based Imputation: Use multiple linear regression if the missing indicator correlates strongly with others.

Imputation models should have an R^2 greater than or equal to 0.7. If R^2 lower than 0.7, consider alternative data sources or methods.

See the example below for how to reproduce the boxplots 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)

Alberto Castro & Axel Luyten

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 healthiar assessments*

Description

This function considers socio-economic aspects (e.g. multiple deprivation index) in the attributable health impacts. 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`) Argument must be entered for iterations. See Details for more info.
`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. Based on this and `n_quantile`, `social_quantile` will be calculated.
`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. See Details for more info.
`pop_fraction`	*(only if* `output_attribute` *not specified)* `Numeric vector` specifying the population attributable fraction by age group and geographic unit.

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

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 = rep(c("below_40", "40_plus"), each = 9037),
    exp_central = c(exdat_socialize$PM25_MEAN, exdat_socialize$PM25_MEAN-0.1),
    cutoff_central = 0,
    rr_central = 1.08,
    erf_shape = "log_linear",
    rr_increment = 10,
   bhd_central =  c(exdat_socialize$MORTALITY_below_40, exdat_socialize$MORTALITY_40_plus),
   population = c(exdat_socialize$POPULATION_below_40, exdat_socialize$POPULATION_40_plus),
    geo_id_micro = rep(exdat_socialize$CS01012020, 2))

## Difference in attributable impacts between geographic units
## that is attributable to differences in deprivation
results <- socialize(
  age_group = c("below_40", "40_plus"),
  ref_prop_pop = c(0.5, 0.5),
  output_attribute = results_age_groups,
  geo_id_micro = exdat_socialize$CS01012020,
  social_indicator = exdat_socialize$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.

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

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$health_detailed$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 determines summary uncertainty (based on central, lower and upper estimates of at least one input variable) using attribute() or compare() function output by 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

Method

For each processed input variable with a provided 95% confidence interval value, a distribution is fitted (see below). From these, `n_sim` input value sets are sampled to compute `n_sim` attributable impacts. The median value of these attributable impacts is reported as the central estimate, and the 2.5th and 97.5th percentiles define the lower and upper bounds of the 95% summary uncertainty confidence interval, respectively. Aggregated central, lower and upper estimates are obtained by summing the corresponding values of each lower level unit.

Distributions used for simulation

Relative risk values are simulated based on an optimized gamma distribution, which fits well as relative risks are positive and its distributions usually right-skewed. The gamma distribution best fitting the inputted central relative risk estimate and corresponding lower and upper 95% confidence interval values is fitted using `stats::qgamma()` (with `rate = shape / rr_central`) and then `stats::optimize` is used to optimize the distribution parameters. Finally, `n_sim` relative risk values are simulated using `stats::rgamma()`.

Exposure values are simulated based on a normal distribution using `stats::rnorm()` with `mean = exp_central` and a standard deviation based on corresponding lower and upper 95% exposure confidence interval values.

Cutoff values are simulated based on a normal distribution using `stats::rnorm()` with `mean = cutoff_central` and a standard deviation based on corresponding lower and upper 95% cutoff confidence interval values.

Baseline health data values are simulated based on a normal distribution using `stats::rnorm()` with `mean = bhd_central` and a standard deviation based on corresponding lower and upper 95% exposure confidence interval values.

Disability weights values of the morbidity health outcome of interest are simulated based on a beta distribution, as both the disability weights and the beta distribution are bounded by 0 and 1. The beta distribution best fitting the inputted central disability weight estimate and corresponding lower and upper 95% confidence interval values is fitted using `stats::qgamma()` (the best fitting distribution parameters `shape1` and `shape2` are determined using `stats::optimize()`). Finally, `n_sim` disability weight values are simulated using `stats::rbeta()`.

Duration values of the morbidity health outcome of interest are simulated based on a normal distribution using `stats::rnorm()` with `mean = duration_central` and a standard deviation based on corresponding lower and upper 95% exposure confidence interval values.

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.

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

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