

healthiar :: CHEAT SHEET

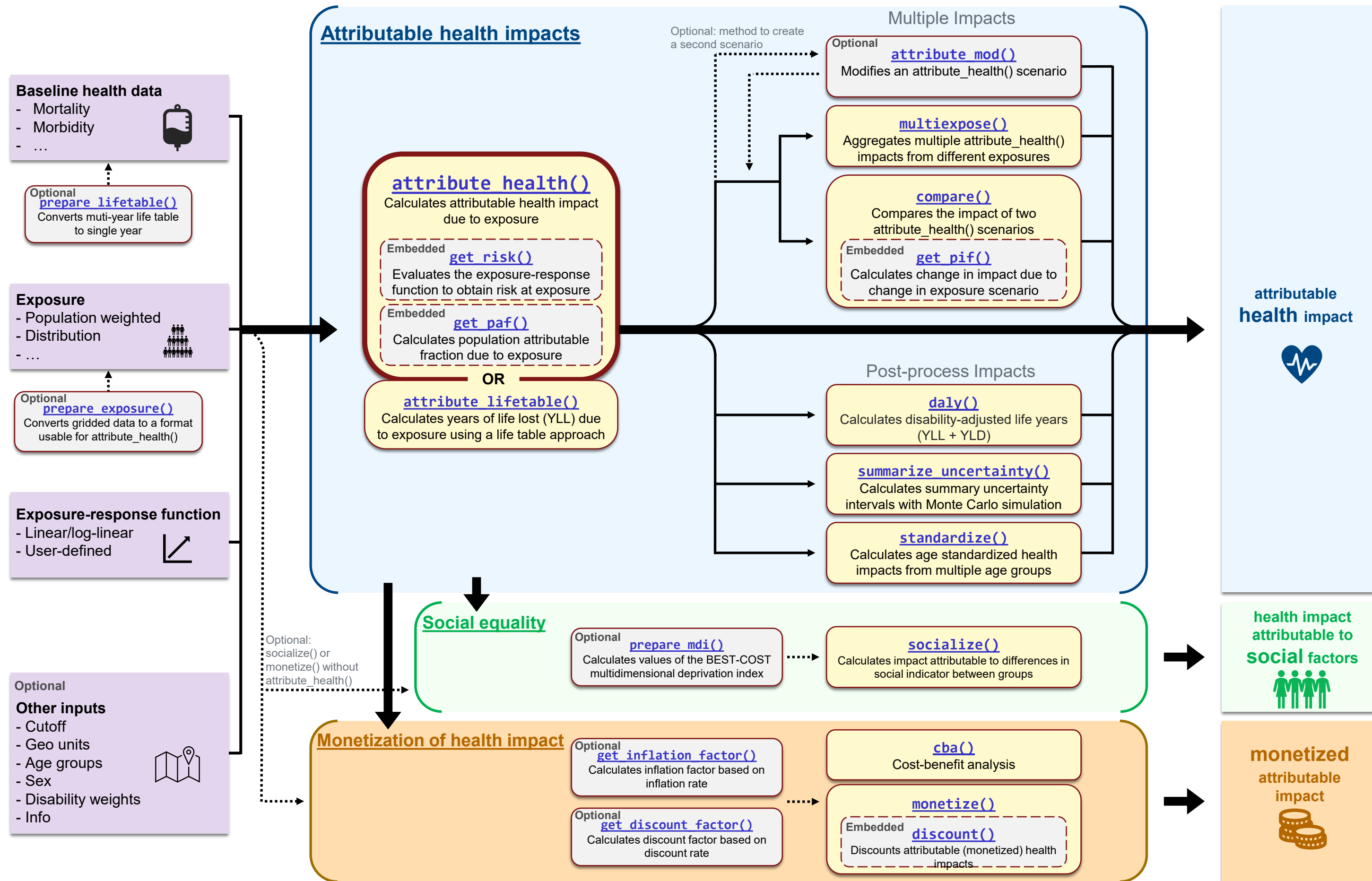


Package overview

Input

Calculation

Output



healthiar : : CHEAT SHEET



General Tipps

Input format of function arguments

Format

Compatible: Single value or vector as numerical(s) or string(s)

Incompatible: Lists, data frames, matrices, tibbles
(Exception: "info" argument)

Structure

For input data **tidy structure** is recommended!

bhd	exp_a	exp_b	exp_c	rr_central	...
1500	13	15	17	1.4	...

OK

Baseline health data (**bhd**), exposure (**exp**), exposure-response function (**erf**)

geo_id	exp	rr_central	bhd	...
"a"	13	1.4	1500	...
"b"	15	1.4	1500	...
"c"	17	1.4	1500	...

BEST- OPTION

Function input (example with attribute_health() function)

Enter input manually

```
results <- attribute_health(  
  exp_central = c(13,15,17),  
  bdh_central = c(1500,1500,1500), #or only 1500  
  erf_shape = "linear",  
  rr_central = c(1.4,1.4,1.4), #or only 1.4  
  rr_increment = c(10,10,10), #or only 10  
  geo_id_micro = c("a", "b", "c"))
```

Enter input from table

```
data <- tidy_table_above1500  
results <- attribute_health(  
  exp_central = data$exp  
  bdh_central = data$bdh  
  erf_shape = data$erf_shape,  
  rr_central = data$rr_central  
  rr_increment = data$increment  
  geo_id_micro = data$geo_id)
```

Output structure

attribute_health() as example

→ Other functions (e.g. **monetize()**, **socialize()**, **compare()**) have similar output structures

Output (nested list)

```
health_main      # main results (central uncertainty path)  
└─ impact        # attributable health impact  
└─ pop_fraction  # population attributable fraction (PAF)  
└─ rr_at_exp     # risk at exposure level  
└─ ...  
health_detailed  # detailed and interim results  
└─ input_args    # raw entered input arguments  
└─ input_table   # input prepared for function  
└─ results_raw   # all results for all uncertainty and group combinations  
└─ results_by_geo_id_macro # results aggregated at macro geo unit(s)  
└─ results_by_geo_id_micro  
└─ ...  
Other possible results tibbles: results_by_sex, results_by_age_group,  
results_by_exp_category,  
results_by_year...
```

The output of **attribute_health()** ... is the attributable health impact
... can be used as input for many other functions in healthiar (see first page)

Find more information on ...

- [Package Website](#)
- [GitHub Repository](#)
- [Vignette](#)
- [CRAN](#)

Pathways – decisions to make

decisions	argument	options
1. Choose risk approach	approach_risk	RR AR
2. Choose exposure type	exp_central	Single Distribution
	erf_eq_central	Formula User-defined function
3. Choose erf	or (only for RR) erf_shape + rr_central + rr_increment	Formula User-defined function Linear Log-linear
Optional extensions		
4. Add cutoff to erf	cutoff_central	Examples 1
5. Set upper and lower bounds	• ..._upper • ..._lower	Examples Value(s) in exp_central +1 Value(s) in exp_central -1
6. Add years lived with disability(YLD)	dw_central	Examples 0.5
7. Geographical analysis	• geo_id_micro • geo_id_macro	Paris Lyon Milan Rome ... France Italy ...
8. Sub-group analysis	• sex • age_group	Female Male 20-30 30-40 ...

Risk approach

Relative Risk (RR)	Absolut Risk (AR)
Required Inputs: bhd: bhd_central exp: exp_central erf: erf_eq_central or erf_shape , rr_central and rr_increment	Required Inputs: bhd: - exp: exp_central , pop_exp erf: erf_eq_central
results <- attribute_health(approach_risk = "relative_risk", exp_central = 8.85, bdh_central = 30747, erf_shape = "log_linear", rr_central = 1.369, rr_increment = 10)	results <- attribute_health(approach_risk = "absolute_risk", exp_central = c(57.5, 62.5), bdh_central = 30747, pop_exp = c(387500, 250000), erf_eq_central = "78-3.1*c+0.1*c^2")

attribute_health()

Exposure type

Single exposure

- One exp value leads to one impact

bhd	exp	erf	rr_central
1500	13	"linear"	1.36

One Impact

Exposure Distribution

- Multiple exp values lead to one impact

Additionally required: **prop_pop_exp**

- Sum = 1
- default = 1 / #exp values (here 4)

→ All inputs except of exp and prop_pop_exp have one fixed value

prop_pop_exp	exp	erf	rr_central	bhd
0.25	12	"linear"	1.36	1500
0.35	13	"linear"	1.36	1500
0.2	15	"linear"	1.36	1500
0.2	17	"linear"	1.36	1500

PAF 1	PAF 2	PAF 3	PAF 4	1500
-------	-------	-------	-------	------

One Impact

Exposure Distribution

- Multiple exposure values lead to multiple impacts (one per exp category)
- pop_exp** → absolute number of exposed people per exposure category

→ All inputs except of exp and pop_exp have one fixed value

pop_exp	exp	erf
3750	12	"70-3*c+0.04*c^2"
5250	13	"70-3*c+0.04*c^2"
3000	15	"70-3*c+0.04*c^2"
3000	17	"70-3*c+0.04*c^2"

Impact 1	Impact 2	Impact 3	Impact 4	One impact per exp category (detailed output)
----------	----------	----------	----------	---

One Impact (main output)

Geographical analysis

Grouping variables: geo_id_micro, geo_id_macro

Grouping:

- Every group needs unambiguous input values
(exception: for exposure distribution exp, pop_exp and prop_pop_exp can be ambiguous)

geo_id_micro	bhd	erf	Prop_pop_exp	rr_central	exp
"paris"	3000 ✗	"linear" ✓	0.7	1.36	13
"paris"	2000 ✗	"linear" ✓	0.3	1.36	15
"rome"	1500 ✓	"log_log" ✗	0.6	1.52	15
"rome"	1500 ✓	"linear" ✗	0.4	1.52	17
	✗	✗	✓	✓	✓

- For every group one single impact is calculated

geo_id_macro	geo_id_micro	bhd	exp	erf	rr_central	results_by_...
"france"	"paris"	3000	13	"linear"	1.36	→ Impact Paris
"france"	"lyon"	2500	15	"linear"	1.54	→ Impact Lyon
"italy"	"rome"	2700	15	"linear"	1.43	→ Impact Rome
"italy"	"milan"	2000	17	"linear"	1.52	→ Impact Milan
						Impact France Impact Italy

Sub-group analysis

Grouping variables: age_group, sex, info

- Grouping works similar as for geo_id_macro and geo_id_micro (see above)
- Sub-groups and geographical analysis grouping can be combined

Example: c("paris", "paris", "rome", "rome") and c("male", "female", "male", "female") =
c("paris.male", "paris.female", "rome.male", "rome.female") = four groups