



Swiss TPH

Hello, I'm healthiar!



Axel Luyten and Alberto Castro

axel.luyten@swisstph.ch & alberto.castrofernandez@swisstph.ch

2025-10-01

Note

This presentation was part of an online *healthiar* workshop by [Swiss TPH](#) held on 2025-10-01, which was recorded. [The recording \(45'\) can be found here:](#)

https://team.swisstph.ch/s/aN_wN5MUTAS3bwEkWvtvaQ

See the README file of the [healthiar GitHub repository](#) for [information on how to install healthiar in RStudio](#) and how to get started: <https://github.com/SwissTPH/healthiar?tab=readme-ov-file#readme>

Feedback is very welcome via [GitHub issues](#) (preferred) or by email (addresses on first slide).

Disclaimer: the R package is still a work in progress - it could be that some of the exact code in this presentation will cease to work at some point, but the concepts behind them should stay the same.

The slides of this presentation can be found here:

https://github.com/SwissTPH/healthiar/tree/master/varia/Workshops_and_demos/workshop

Today

First hour

- Hello, I'm `healthiar`!
- `healthiar` function examples
- Outlook
- Q & A

Break

Second hour

- Hands-on mock case studies
- Q & A

Hello, I'm healthiar!

healthiar in numbers

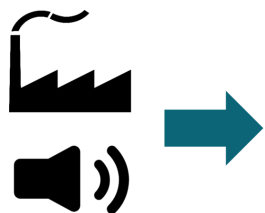
- **1** purpose - to help you quantify and monetize the attributable burden of disease
- **3** dimensions - health, monetization, social aspects
- **20** functions
- **> 1000** calculation pathways
- **60000** (simple) assessments per second

healthiar overview

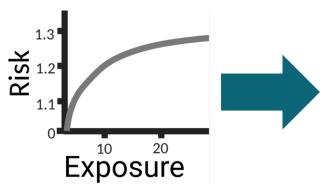
Baseline
health
data



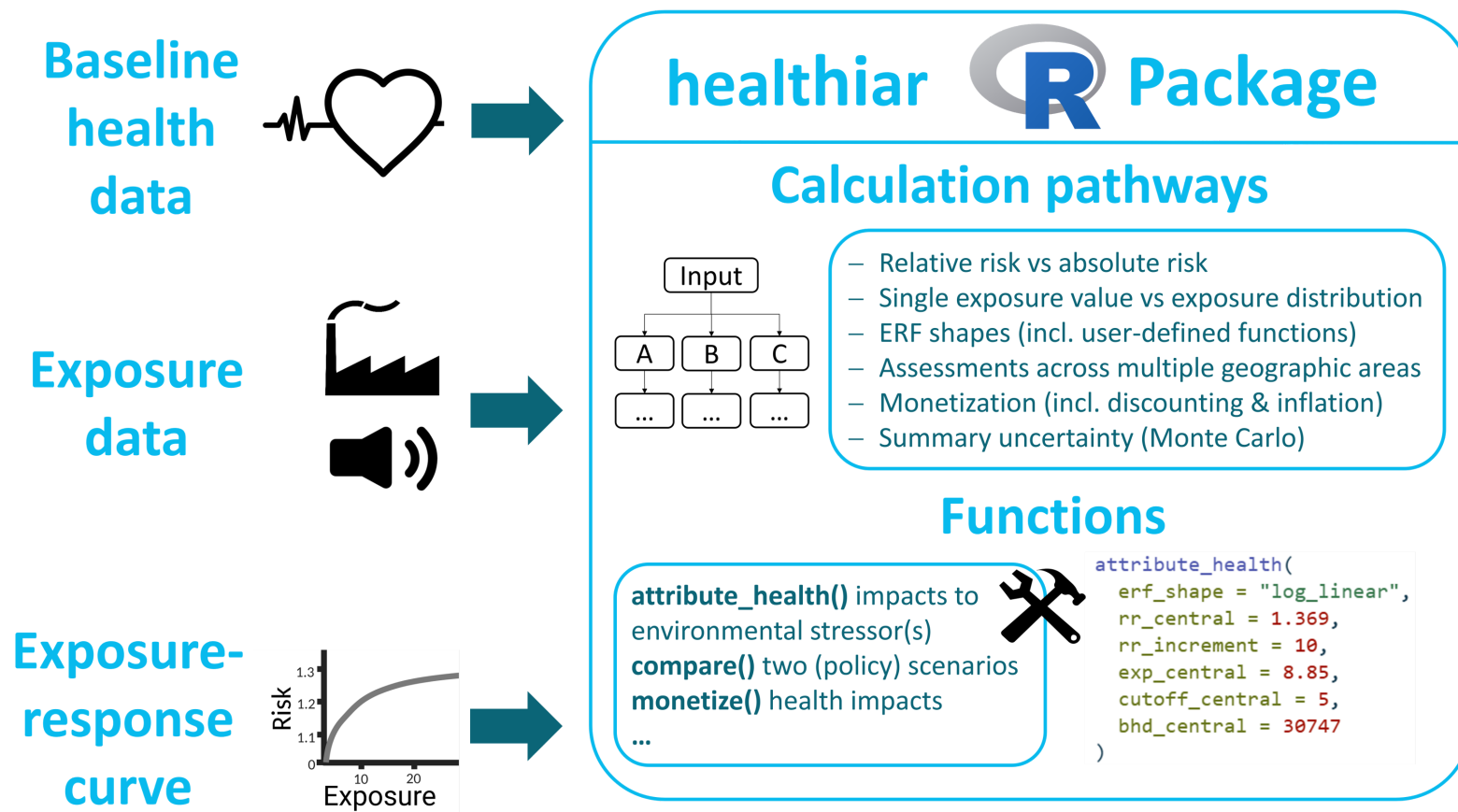
Exposure
data



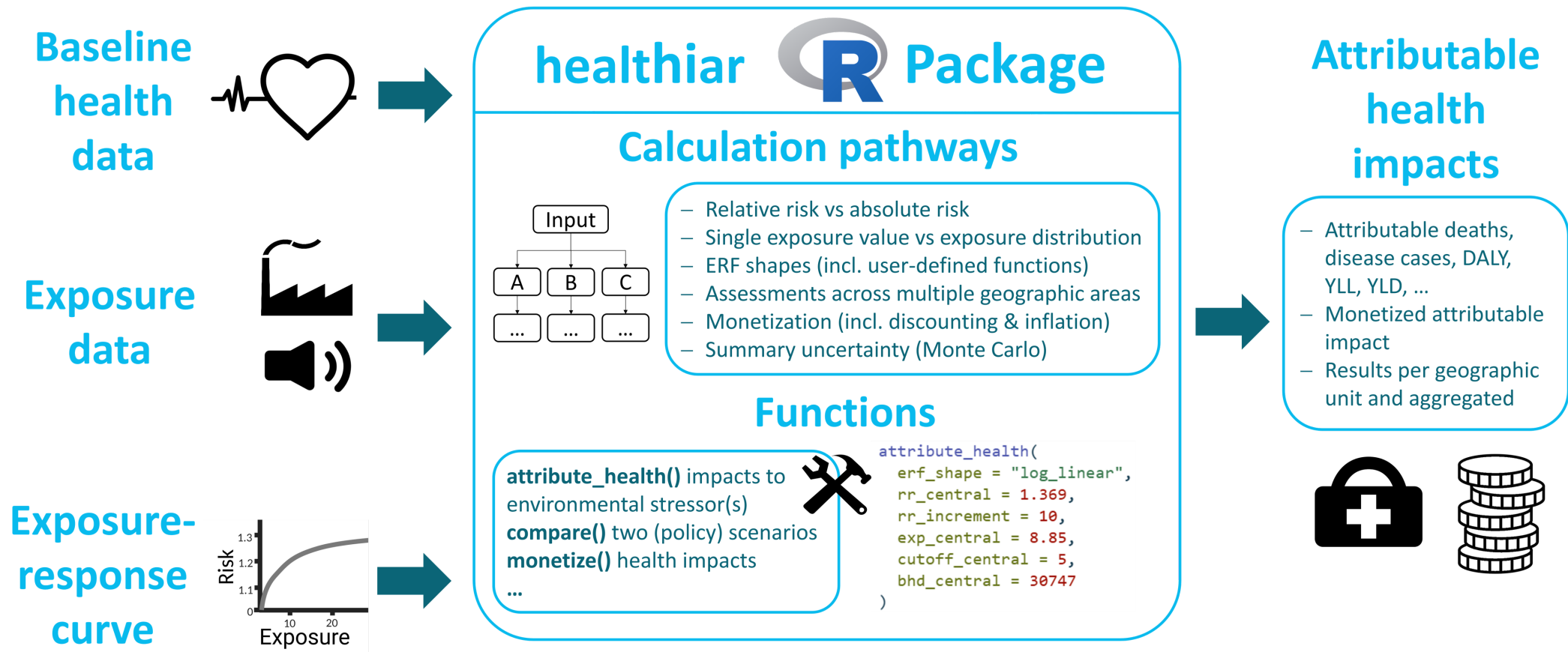
Exposure-
response
curve



healthiar overview

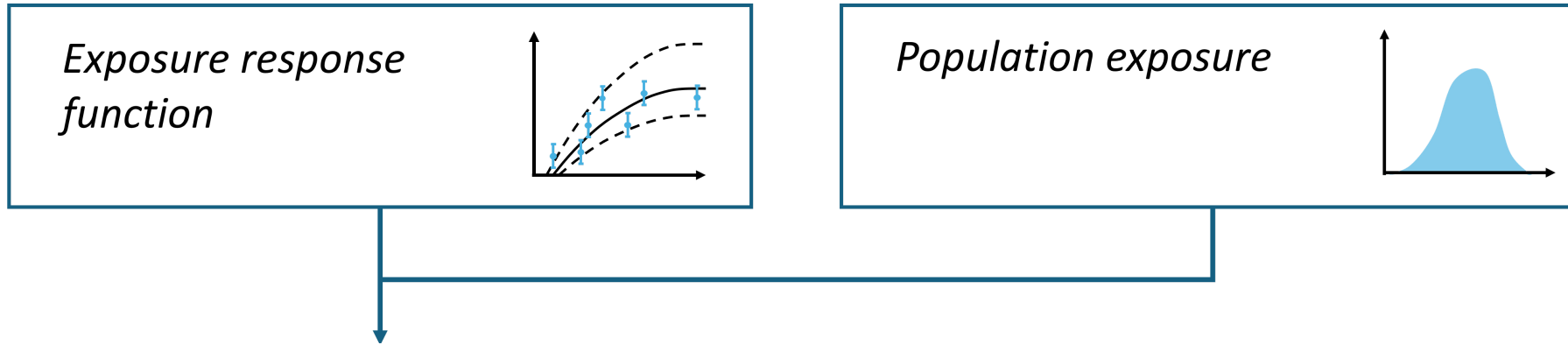


healthiar overview

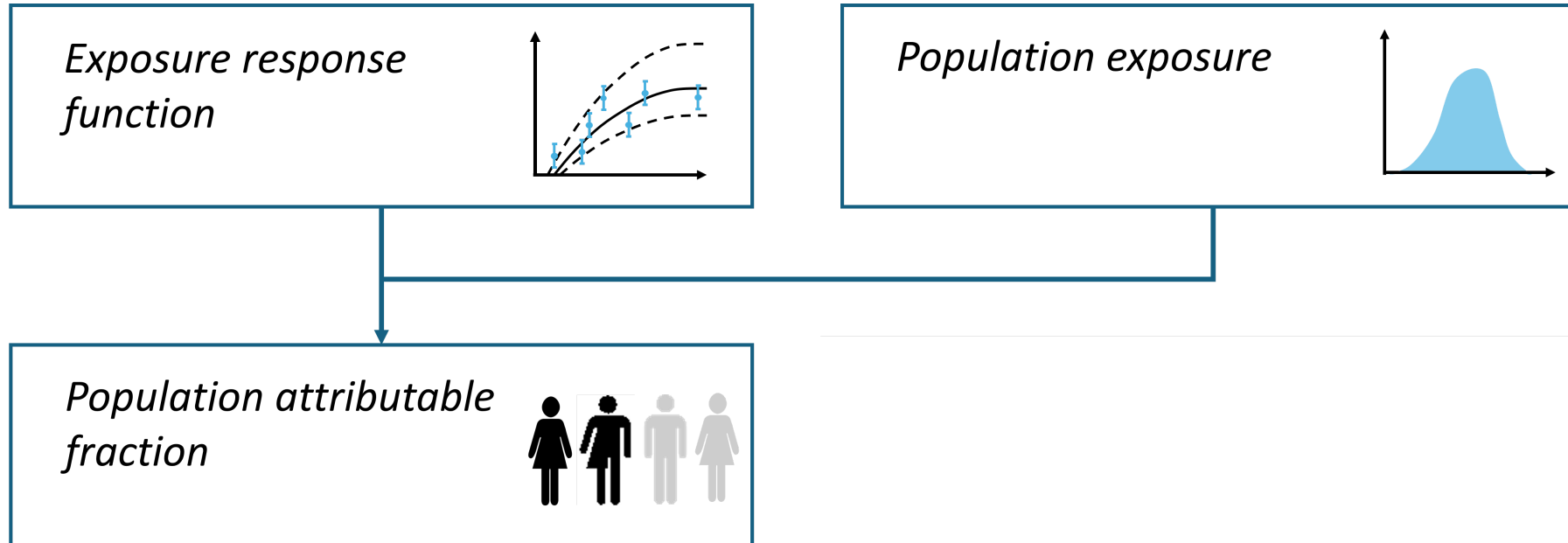


Example:
attribute_health()
with RR

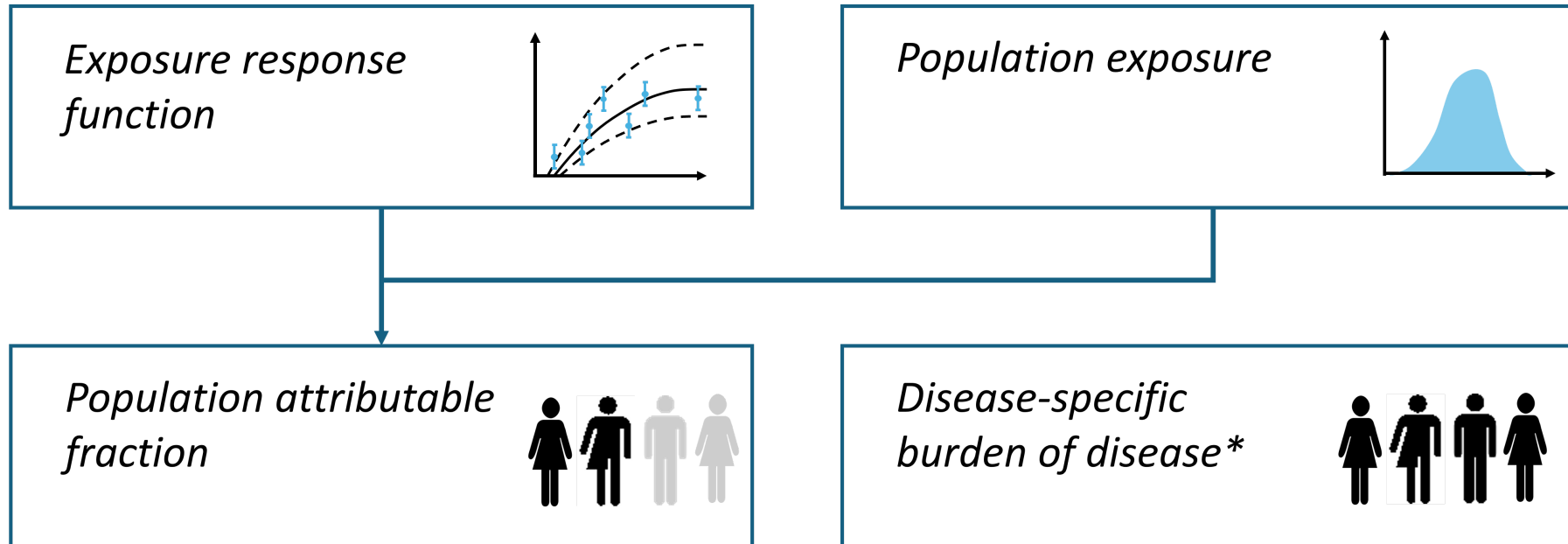
Refresher - Burden of disease with relative risk



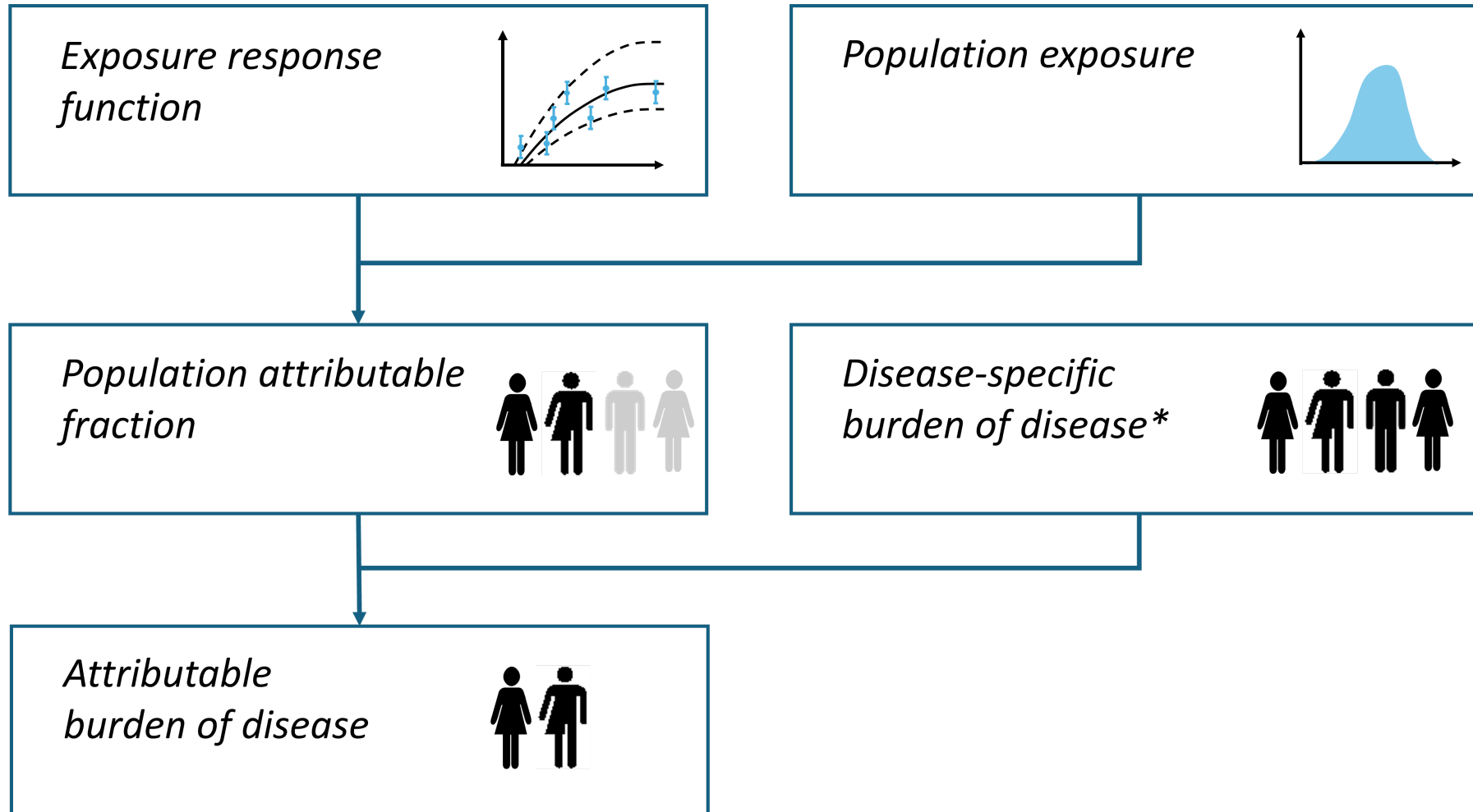
Refresher - Burden of disease with relative risk



Refresher - Burden of disease with relative risk



Refresher - Burden of disease with relative risk



attribute_health() with RR 1/3

Goal: attribute COPD cases to PM2.5 air pollution exposure

Hard-coded input data

Pre-loaded data

```
1 results_pm_copd <- attribute_health(  
2   erf_shape = "log_linear", # erf = exposure-response function  
3   rr_central = 1.369,  
4   rr_increment = 10, # µg / m^3  
5   exp_central = 8.85, # µg / m^3  
6   cutoff_central = 5, # µg / m^3  
7   bhd_central = 30747 # bhd = baseline health data (here: COPD incidence)  
8 )
```

attribute_health() with RR 2/3



Output structure

`attribute_health()` outputs two lists (“folders”)

- `health_main` contains the main results
- `health_detailed` contains more detailed results and other assessment details in sub-lists (“sub-folders”)

attribute_health() with RR 3/3

With \$ operator

[View\(...\)](#)

By clicking

```
1 results_pm_copd$health_main
```

impact_rounded	impact	pop_fraction	erf_ci	rr	exp	bhd
3502	3501.962	0.1138961	central	1.369	8.85	30747



Interpretation

3'502 cases can be attributed to PM2.5 exposure.

Example: **compare()** two
scenarios

compare() two scenarios 1/2

Goal: compare attributable health impacts in two (policy) scenarios

1. Use `attribute_health()` to calculate burden of scenarios 1 & 2

```
1 scenario_1 <- attribute_health(  
2   erf_shape = "log_linear",  
3   rr_central = 1.369,  
4   rr_increment = 10,  
5   exp_central = 8.85, # SCEN 1  
6   bhd_central = 30747  
7 )
```

```
1 scenario_2 <- attribute_health(  
2   erf_shape = "log_linear",  
3   rr_central = 1.369,  
4   rr_increment = 10,  
5   exp_central = 6, # SCEN 2  
6   bhd_central = 30747  
7 )
```

2. Use `compare()` to compare scenarios 1 & 2

```
1 results_comparison <- compare(  
2   output_attribute_scen_1 = scenario_1,  
3   output_attribute_scen_2 = scenario_2  
4 )
```

compare() two scenarios 2/2

```
1 results_comparison$health_main
```

impact	impact_rounded	impact_scen_1	impact_scen_2	bhd	exp_category
2180.49	2180	7461.493	5281.003	30747	1

Interpretation

By implementing the clean air policy we expect that 2'180 COPD cases can be avoided.

Example:
attribute_health()
across multiple
geographic units

attribute_health() across multiple geographic units 1/3

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

```
1 results_iteration <- attribute_health(  
2   geo_id_micro = c("Zürich", "Basel", "Geneva", "Ticino", "Jura"),  
3   geo_id_macro = c("German", "German", "French", "Italian", "French"),  
4   exp_central = c(11, 11, 10, 8, 7),  
5   bhd_central = c(4000, 2500, 3000, 1500, 500),  
6   erf_shape = "log_linear",  
7   rr_central = 1.369,  
8   rr_increment = 10,  
9   cutoff_central = 5  
10 )
```

attribute_health() across multiple geographic units 2/3

```
1 results_iteration$health_main
```

geo_id_macro	impact_rounded	erf_ci	exp_ci	bhd_ci
German	1116	central	central	central
French	466	central	central	central
Italian	135	central	central	central



Tip

The main results contain results aggregated by `geo_id_macro` (if available).

attribute_health() across multiple geographic units 3/3

```
1 results_iteration$health_detailed$results_raw
```

geo_id_micro	geo_id_macro	impact_rounded
Zürich	German	687
Basel	German	429
Geneva	French	436
Ticino	Italian	135
Jura	French	30



Tip

Impact per geo unit specified in `geo_id_micro` is stored in the detailed results.

Example:

summarize_uncertainty()

summarize_uncertainty() 1/3

If we add 95% confidence intervals to `attribute_health()` ...

```
1 results_pm_copd <- attribute_health(
2   erf_shape = "log_linear",
3   rr_central = 1.369, rr_lower = 1.114, rr_upper = 1.664,
4   rr_increment = 10,
5   exp_central = 8.85, exp_lower = 8, exp_upper = 10,
6   cutoff_central = 5,
7   bhd_central = 30747, bhd_lower = 28000, bhd_upper = 32000
8 )
```

... then the folder `health_detailed` contains all different uncertainty combinations:

```
1 results_pm_copd$health_detailed$results_raw
```

erf_ci	exp_ci	bhd_ci	impact_rounded	rr	exp	bhd
central	central	central	3502	1.369	8.85	30747
lower	central	central	1252	1.114	8.85	30747
upper	central	central	5474	1.664	8.85	30747

erf_ci	exp_ci	bhd_ci	impact_rounded	rr	exp	bhd
central	central	lower	3189	1.369	8.85	28000
lower	central	lower	1140	1.114	8.85	28000
upper	central	lower	4985	1.664	8.85	28000
central	central	upper	3645	1.369	8.85	32000
lower	central	upper	1303	1.114	8.85	32000
upper	central	upper	5697	1.664	8.85	32000
central	lower	central	2765	1.369	8.00	30747
lower	lower	central	980	1.114	8.00	30747
upper	lower	central	4356	1.664	8.00	30747
central	lower	lower	2518	1.369	8.00	28000
lower	lower	lower	892	1.114	8.00	28000
upper	lower	lower	3967	1.664	8.00	28000
central	lower	upper	2877	1.369	8.00	32000
lower	lower	upper	1020	1.114	8.00	32000

erf_ci	exp_ci	bhd_ci	impact_rounded	rr	exp	bhd
upper	lower	upper	4533	1.664	8.00	32000
central	upper	central	4468	1.369	10.00	30747
lower	upper	central	1616	1.114	10.00	30747
upper	upper	central	6911	1.664	10.00	30747
central	upper	lower	4069	1.369	10.00	28000
lower	upper	lower	1471	1.114	10.00	28000
upper	upper	lower	6294	1.664	10.00	28000
central	upper	upper	4651	1.369	10.00	32000
lower	upper	upper	1682	1.114	10.00	32000
upper	upper	upper	7193	1.664	10.00	32000

summarize_uncertainty() 2/3

Goal: perform a Monte Carlo simulation to obtain summary uncertainty confidence intervals that combine uncertainty in different input data

Step 1: Use `attribute_health()` with uncertainty in ≥ 2 function arguments

```
1 results <- attribute_health(  
2   erf_shape = "log_linear",  
3   rr_central = 1.369,  
4   rr_lower = 1.114,  
5   rr_upper = 1.664,  
6   rr_increment = 10,  
7   exp_central = 8.85,  
8   exp_lower = 8,  
9   exp_upper = 10,  
10  cutoff_central = 5,  
11  bhd_central = 30747,  
12  bhd_lower = 28000,  
13  bhd_upper = 32000)
```

Step 2: `summarize_uncertainty()` to obtain the confidence intervals

```
1 results <-  
2   summarize_uncertainty(  
3     output_attribute = results,  
4     n_sim = 100  
5   )
```

summarize_uncertainty() 3/3

```
1 results$uncertainty_main
```

impact_ci	impact	impact_rounded
central_estimate	3136.993	3137
lower_estimate	1537.385	1537
upper_estimate	5676.227	5676

Example:
attribute_health()
with RR and user-
defined ERF

attribute_health() with RR and user-defined ERF 1/2

Goal: attribute COPD cases to air pollution exposure by applying a user-defined exposure response function (e.g. MR-BRT curves from Global Burden of Disease study)

- Any function of the form $intercept + a \times c^1 + b \times c^2 + \dots$
- Any other (non-linear) function of the `function` type, as obtained from e.g. `splinefun()` or `approxfun()`

```
1 results_mr_brt <- attribute_health(  
2   exp_central = 8.85,  
3   bhd_central = 30747,  
4   erf_eq_central = splinefun(  
5     x = c(0, 5, 10, 15, 20, 25, 30, 50, 70, 90, 110),  
6     y = c(1.00, 1.04, 1.08, 1.12, 1.16, 1.20, 1.23, 1.35, 1.45, 1.53, 1.60),  
7     method = "natural")  
8 )
```

attribute_health() with RR and user-defined ERF 2/2

```
1 results_mr_brt$health_main
```

impact_rounded	impact	pop_fraction	erf_ci	rr_at_exp	exp	bhd
2033	2032.583	0.0661067	central	1.070786	8.85	30747

Example:
attribute_health()
with RR and exposure
distribution

attribute_health() with RR and exposure distribution

Goal: attribute COPD cases to air pollution exposure categories

```
1 results_pm_copd <- attribute_health(
2   erf_shape = "log_linear",
3   rr_central = 1.369,
4   rr_increment = 10,
5   exp_central = c(5, 6, 7, 8, 9, 10),
6   prop_pop_exp = c(0.2, 0.1, 0.2, 0.3, 0.1, 0.1),
7   cutoff_central = 5,
8   bhd_central = 30747
9 )
```

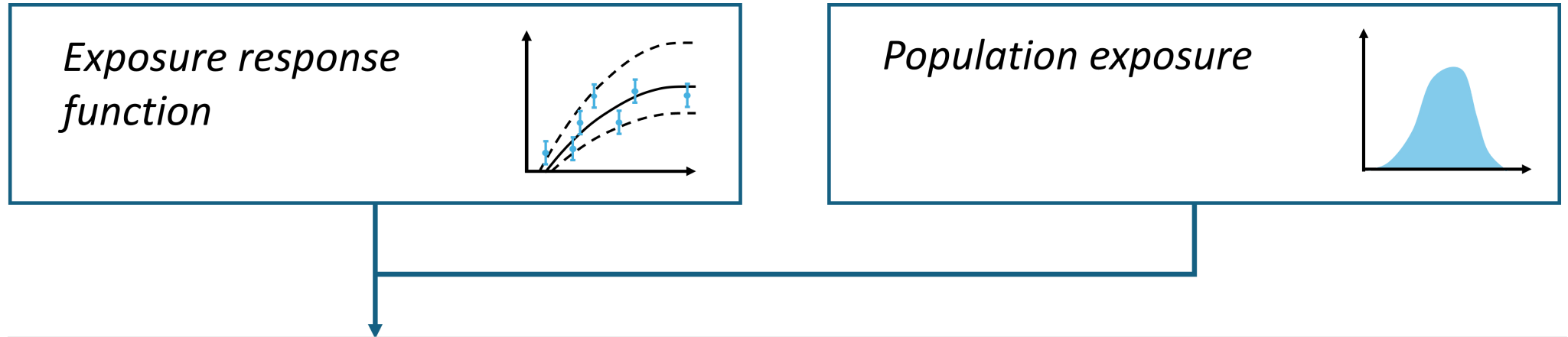
```
1 results_pm_copd$health_main
```

impact_rounded	pop_fraction	exp	prop_pop_exp	rr_at_exp
2177	0.0707956	5, 6,	0.2, 0.1, 0.2,	1, 1.03190649221009,
		7, 8,	0.3, 0.1, 0.1	1.06483100866533,
				1.09880603094837,

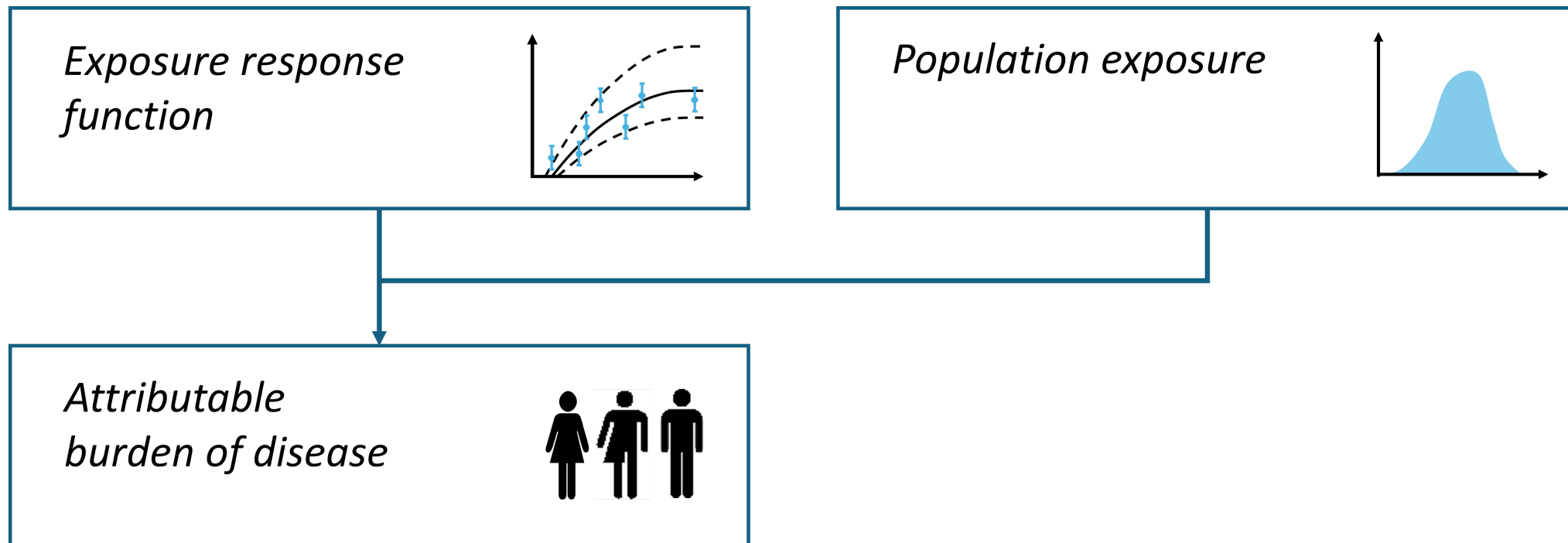
impact_rounded	pop_fraction	exp	prop_pop_exp	rr_at_exp
		9,		1.13386507701522,
		10		1.1700427342623

Example:
attribute_health()
with AR

Refresher - Burden of disease with absolute risk



Refresher - Burden of disease with absolute risk



attribute_health() with AR 1/2

Goal: attribute cases of high annoyance (HA) to noise exposure

Exposure category	Exposure mean	Population exposed
exp < 55		4'268'785
55 ≤ exp < 60	57.5	387'500
60 ≤ exp < 65	62.5	286'000
65 ≤ exp < 70	67.5	191'800
70 ≤ exp < 75	72.5	72'200
75 ≤ exp	77.5	7'700

```

1 results_noise_ha <- attribute_health(
2   approach_risk = "absolute_risk",
3   erf_eq_central = "78.9270-3.1162*c+0.0342*c^2",
4   exp_central = c(57.5, 62.5, 67.5, 72.5, 77.5),
5   pop_exp = c(387500, 286000, 191800, 72200, 7700),
6 )

```

attribute_health() with AR 2/2

```
1 results_noise_ha$health_main
```

approach_risk	exp_category	impact
absolute_risk	1, 2, 3, 4, 5	174231.8

```
1 results_noise_ha$health_detailed$results_raw
```

exp_category	exp	pop_exp	impact
1	57.5	387500	49674.594
2	62.5	286000	50788.595
3	67.5	191800	46813.105
4	72.5	72200	23657.232
5	77.5	7700	3298.314

Other healthiar functions

Other healthiar functions

- `attribute_lifetable()` for YLL
- `attribute_health()` for YLD (using the `dw` & `duration` arguments)
- `daly()` sums attributable YLL and YLD impacts
- `monetize()` monetizes (attributable) health impacts
- `cba()` performs a cost-benefit analysis (CBA)
- `multiexpose()` considers exposure to 2 exposures (e.g. two different air pollutants) at the same time to quantify the attributable health impacts
- `prepare_exposure()` calculates (population-weighted) mean exposure by combining spatial exposure and geographic units data
- `socialize()` determines burden attributable to differences in a social indicator
- `attribute_mod()` modifies an existing `attribute_health()` assessment
- `prepare_mdi()` creates the BEST-COST MDI (Multidimensional Deprivation Index)

Outlook

Outlook

- [healthiar](#) is publicly available on this [GitHub page](#)
- Package submitted to CRAN
- Python version coming out later in 2026
- [healthiar](#) might be expanded in the future (if funding available)

Q & A

Second hour programme

Second hour programme

- Exercises in small groups (ca. 40 min)
- Presentation of solutions in plenum (ca. 10 min)
- Q & A (remaining time)

Exercises and **slides** from this presentation can be downloaded from [this folder on the healthiar GitHub page](#) (link also in the last email and in the Zoom chat).

Install and load the `healthiar` R package (code also found at the start of the exercises).

If asked to, please install or update dependencies.

```
1 install.packages(c("remotes", "knitr", "rmarkdown"))
2 remotes::install_github(repo = "SwissTPH/healthiar", build_vignettes = TRUE)
3 library(healthiar)
```

Happy coding!

Q & A

Feedback

Feedback

Please answer 3 short questions to provide your valuable feedback via [this short form](#) (link also in the last email and in the Zoom chat)

Also, any input via [GitHub issues](#) is also welcome.

Or you can write to Alberto (alberto.castrofernandez@swisstph.ch) and Axel (axel.luyten@swisstph.ch).

Appendix

Additional function examples

For additional function examples please see the function documentations and the package vignette.

The next slides will show you how to access them.

healthiar in RStudio 1/3

R package healthiar to quantify and monetize the burden of disease attributable to exposures



Documentation for package 'healthiar' version 0.0.2.832

- [DESCRIPTION file.](#)
- [User guides, package vignettes and other documentation.](#) Find here the intro vignette
- [Package NEWS.](#)

Help Pages = Function documentations

[attribute_health](#)

Attribute health impacts to an environmental stressor

...

...

Figure: Landing page of the [healthiar](#) package in RStudio, where you find the package vignettes and function documentation.

healthiar in RStudio 2/3

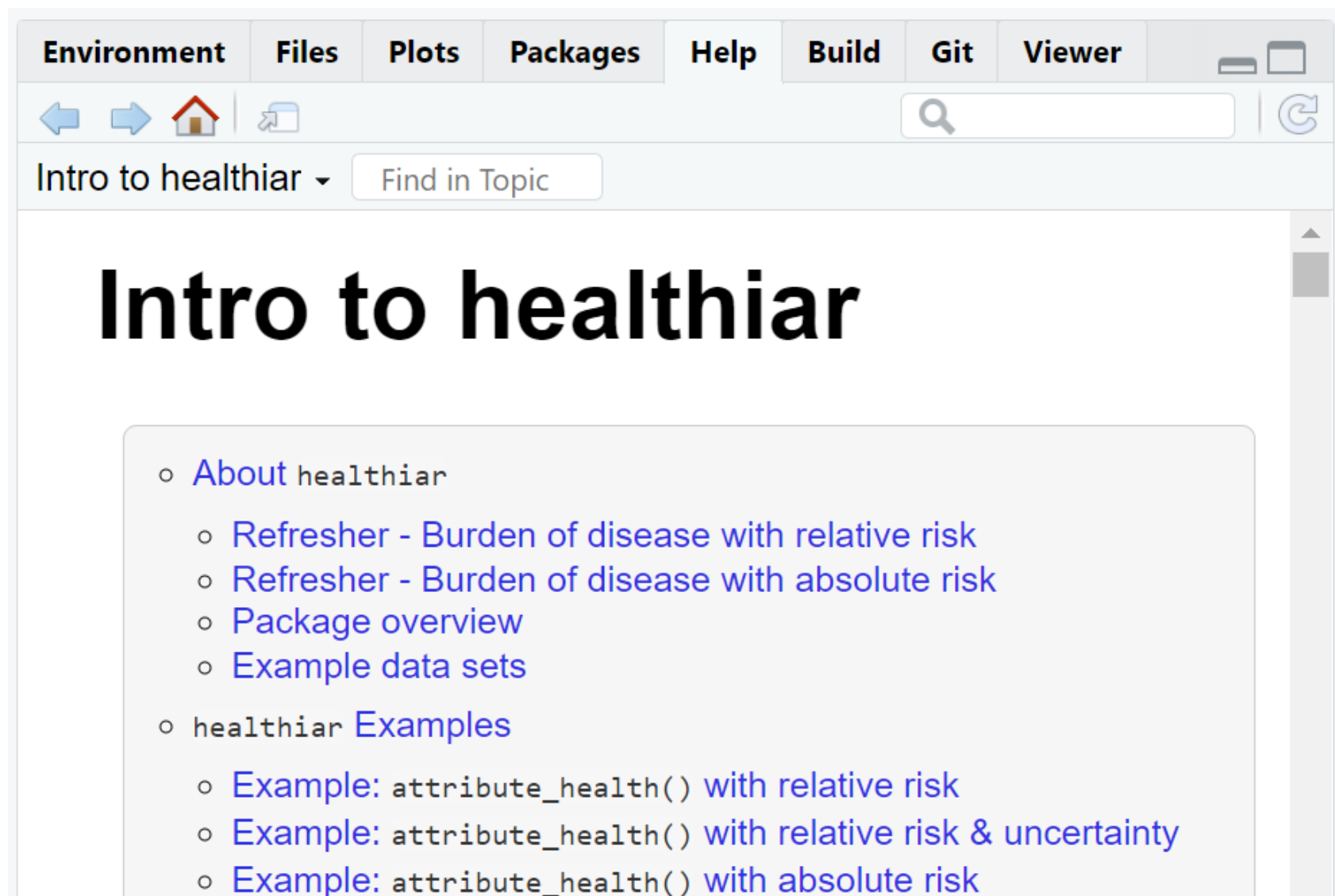


Figure: Intro to [healthiar](#) vignette (= detailed guide to the package)

healthiar in RStudio 3/3

attribute_health {healthiar}

(preview) R Documentation

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

Figure: Help page of `attribute_health()`

Export results

Export as .csv file

Save as .Rdata file

Export to Excel (as .xlsx file)

```
1 write.csv(x = results_pm_copd$health_main, file = "exported_results/results_pm
```

Visualize results

Visualization is out of scope of [healthiar](#). You can visualize in

- R, e.g. with the [ggplot2](#) package ([online book by the creator](#))
- Excel (export results first)
- Other tools

