Example_adding_a_test

Axel Luyten

Example: Adding a test

1. Pick a pathway

You want to make sure the results for the following calculation pathway are correct: - Relative risk - ERF shape: log-linear - Exposure: exposure distribution - With cutoff - No iteration - No input variable uncertainty

2. Look for comparison assessment

You remember that you've just assessed such a case recently! You dig up the assessment, both the input data and the results, and get set to add a test.

3. Fill out the function template with the assessment input data

You select the appropriate template that Swiss TPH provided. You make sure that the package "testthat" (more info: https://testthat.r-lib.org/) is installed.

If you haven't already, install & load the most recent version of the BEST-COST R package healthiar following the instructions in the file "intro_to_healthiar.html", located in the same teams folder as this file.

```r
testthat::test_that("results correct, YOUR NAME (TEST_ID)", {

  ## IF APPLICABLE: LOAD INPUT DATA BEFORE RUNNING THE FUNCTION
  # data <- ...

  testthat::expect_equal(
    ## healthiar FUNCTION CALL
    object =
      healthiar::attribute_health(
        exp_central = ,
        prop_pop_exp = ,
        cutoff_central = ,
        bhd_central = ,
        rr_central = ,
        rr_increment = ,
        erf_shape =
      ) |>
      healthiar::helper_extract_main_results(),
    ##  RESULT(S) FROM THE COMPARISON ASSESSMENT YOU SELECTED
    expected =
      c( )
    )
})

## Assessment details: Add here short description of the assessment: year, metric (e.g. DALY, premature
## Input data details: Add here input data details: data sources, measured vs. modelled, ...
```

Then you fill out the template using the input data of your selected assessment, make sure you have loaded the healthiar package and run the test.

NOTE: depending on the assessment, it might be easier to load your needed input data, just before the healthiar function call (see template). In that case, please also upload the input data to the Teams testing folder of your institute, alongside the code for the test. If you don't have much R programming experience, it's easiest to create the test with hard-coded inputs.

You enter your name, institute abbreviation and pathway ID to the first argument of the `testthat::test_that` call. - Name: Axel Luyten - Institute abbreviation: Swiss TPH - Pathway ID: pw_exp_dist pw_cutoff_TRUE pw_varuncer_FALSE pw_erf_log_lin pw_iteration_FALSE pw_multiexp_FALSE

Test with hard-coded data

Here's the filled out template with hard-coded input data (e.g. no external input data loaded).

```
testthat::test_that("results correct, Axel Luyten, Swiss TPH (pw_exp_dist pw_cutoff_TRUE pw_varuncer_FA

  testthat::expect_equal(
    ## healthiar FUNCTION CALL
    object =
      healthiar::attribute_health(
        exp_central = c(53.0, 57.5, 62.5, 67.5, 72.5, 77.5),
        prop_pop_exp = c(0.818718312, 0.074319355, 0.054852478, 0.036785683, 0.013847374, 0.001476797),
        cutoff_central = 53,
        bhd_central = 85362.08,
        rr_central = 1.08,
        rr_increment = 10,
        erf_shape = "log_linear"
      ) |>
      healthiar::helper_extract_main_results(),
    ##  RESULT(S) FROM THE COMPARISON ASSESSMENT YOU SELECTED
    expected =
      c(1151)
    )
})
```

Test with loaded data

```
testthat::test_that("results correct, Axel Luyten, Swiss TPH (pw_exp_dist pw_cutoff_TRUE pw_varuncer_FA

  ## IF APPLICABLE: LOAD INPUT DATA BEFORE RUNNING THE FUNCTION
  data_raw  <-  readxl::read_xlsx(path = "../testing/input/noise_niph/example_road_noise_niph.xlsx",
  # data_raw  <-  readxl::read_xlsx(path = "example_road_noise_niph.xlsx",
                    sheet = "Relative_risk_IHD_WHO_2003a")
  data  <- data_raw |>
    dplyr::filter(!is.na(data_raw$exposure_mean))

  testthat::expect_equal(
    ## healthiar FUNCTION CALL
    object =
      healthiar::attribute_health(
        exp_central = data$exposure_mean,
        prop_pop_exp = data$prop_exposed,
        cutoff_central = min(data$exposure_mean),
        bhd_central = data$gbd_daly[1],
        rr_central = 1.08,
        rr_increment = 10,
        erf_shape = "log_linear") |>
      healthiar::helper_extract_main_results(),
    ##  RESULT(S) FROM THE COMPARISON ASSESSMENT YOU SELECTED
    expected =
      data_raw |>
      dplyr::filter(exposure_category %in% "Total exposed")|>
      dplyr::select(daly)|>
      dplyr::pull() |>
      round()
    )
})
```

4. Check whether results of healthiar and your chosen assessment are the same

```
testthat::test_that("results correct, Axel Luyten, Swiss TPH (pw_exp_dist pw_cutoff_TRUE pw_varuncer_FA

  testthat::expect_equal(
    ## healthiar FUNCTION CALL
    object =
      healthiar::attribute_health(
        exp_central = c(53.0, 57.5, 62.5, 67.5, 72.5, 77.5),
        prop_pop_exp = c(0.818718312, 0.074319355, 0.054852478, 0.036785683, 0.013847374, 0.001476797),
        cutoff_central = 53,
        bhd_central = 85362.08,
        rr_central = 1.08,
        rr_increment = 10,
        erf_shape = "log_linear"
      ) |>
      healthiar::helper_extract_main_results(),
    ##  RESULT(S) FROM THE COMPARISON ASSESSMENT YOU SELECTED
    expected =
      c(1151)
    )
})
```

Yes, they are the same.

5. Upload the test

You upload your test with the file name . . . .