Package: jenner (via r-universe)

September 12, 2024

Title Internal Montagu Helpers

Version 0.0.26

Description Helpers for Montagu.

License MIT + file LICENSE

Author Rich FitzJohn

Maintainer Rich FitzJohn <rich.fitzjohn@gmail.com>

Imports DBI, RcppRoll, RPostgres, vaultr (>= 0.2.0), whisker, yaml

RoxygenNote 6.1.1

Suggests RSQLite, testthat (>= 1.0.2)

Encoding UTF-8

Repository https://vimc.r-universe.dev

RemoteUrl https://github.com/vimc/jenner

RemoteRef master

RemoteSha 5d18fc94b607f95549c9eb7791b12af72efed94f

Contents

admin_set_active_touchstone *Set the active touchstone*

Description

Set the active touchstone

Usage

```
admin_set_active_touchstone(con, touchstone_id, dry_run = TRUE)
```

Arguments

con	Database connection (will require write access to the database)
touchstone_id	Touchstone id to set as "open"
dry_run	Don't commit the transaction - just test if it would work

calculate_dalys *DALYs calculation*

Description

Calculate dalys

Usage

```
calculate_dalys(con, touchstone_name, year_min = 2000, year_max = 2030,
  vimc_dalys_only = TRUE, modelling_group = NULL,
  stochastic_data = NULL, dalys_parameters = NULL, life_table = NULL)

create_dalys_parameters(con, touchstone_name = "201710gavi",
  vimc_dalys_only)

create_dalys_life_table(con, touchstone_name = "201710gavi",
  year_min = 2000, year_max = 2030)
```

Arguments

con	You can be `readonly` user to run this function. But if you need to import dalys for Ferrari, Li and LiST (201710gavi), you can use `import` user.
touchstone_name	touchstone for which dalys are calcualted
year_min	minimal year

`year_max` maximum year

`vimc_dalys_only`
set to be TRUE if we are only interested in Ferrari, Li and LiST (201710gavi)

`modelling_group`
This parameter makes the calcualtion more flexible, specify a vector of modelling_group(s) that you are interested in.

`stochastic_data`
If this is not NULL, then instead of querying the database for the burden estimate set, use the specified stochastic_data, which must be a data frame containing the columns "burden_estimate_set", "country", "year", "age", "burden_outcome" and "burden". The burden_estimate_set should refer to the central burden estimate set for that group, which daly parameters related to. "country" is the 3-character representation. "year" and "age" are trivial; "burden_outcome" is the integer code for the burden, for each line, and "burden" is the data value.

`dalys_parameters`
Leave as NULL, for single calls to calculate_dalys, but for many calls, for example stochastic runs, call create_dalys_parameters first, and pass the result as an argument here to speed things up.

`life_table` Leave as NULL, for single calls to calculate_dalys, but for many calls, for example stochastic runs, call create_dalys_life_table first, and pass the result as an argument here to speed things up significantly.

`create_touchstone` *Create a new touchstone*

Description

Create and import a new touchstone.

Usage

```
create_touchstone(con, dat, demography_from = NULL, path_meta = "meta",
  transaction = TRUE, dry_run = TRUE)
```

Arguments

`con` Database connection. You will need to be the `vimc` or `import` user (not `readonly`) to run this function.

`dat` A `data.frame` of coverage data to import.

`demography_from`
Touchstone id to import demographic statistics from. This must currently be given, but in future we'll allow this to be imported from a csv

`path_meta` Path that we look for various metadata files. Eventually we'll document what they look like.

`transaction` Do in one transaction (logical scalar)

`dry_run` Don't commit (just rollback the transaction after completion)

database_connection *Connect to database*

Description

Connect to database

Usage

```
database_connection(location = "science", user = "readonly",
  local_port = NULL, local_password_group = "science")
```

Arguments

location	One of "science", "production", "uat" or "localhost". Be *very* careful if using production
user	Username to connect as
local_port	Port (when running locally)
local_password_group	Password group (when running locally)

fix_coverage_fvps *Impact Calculation (method 2)*

Description

Provide age-specific coverage-un_pop-fvps

Usage

```
fix_coverage_fvps(con, touchstone_name = "201710gavi", year_min = 2000,
  year_max = 2100, pine = FALSE, write_table = TRUE,
  report_suspecious_coverage = FALSE, touchstone_pop = NULL,
  gavi_support_levels = c("with", "bestminus"))
```

Arguments

con	Database connection. You will need to be `readonly` user to run this function.
touchstone_name	Specify touchstone name only, not with specific version.
year_min	min year of vaccination
year_max	max year of vaccination
pine	this is for testthat. we only grab data for pine countries if true

`write_table` If true, create a temporary table; otherwise return a dataframe

`report_suspecious_coverage` switch on/off the reporting of suspecious coverage

`touchstone_pop` population touchstone, this is for the modups where fvps are calculated using a coverage touchstone and a population touchstone

`gavi_support_levels` specify gavi_support_levels that apply to the touchstone_name you are looking at

---

`impact_calculation` *Impact Calculation (method 2)*

Description

Calcualte impact with method2 that allocate impact by fvps*impact_rate

Usage

```
impact_calculation(con, meta, year_min = 2000, year_max = 2030,
  routine_tot_rate_shape = "trace_cohort", method = "method2",
  age_max = 100)
```

Arguments

`con` Database connection. You will need to be `readonly` user to run this function.

`meta` This is the metadata that goes into the calcualtion

`year_min` minimal year of vaccination

`year_max` maximal year year of vaccination

`routine_tot_rate_shape` This parameter determines how we chop off the year-age matrix to calculate impact rates campaign is stratiforward, use all fvps and all burden estimates to calculate impact rate. So no need to specify. Becuase all impacts (years 2000-2100) are derived from campaigns between 2000 and 2030. Routine is more complicated. We either trance birth cohort between 2000-2030 or trance all birth cohorts between 2000-2100.

`method` impact calculation method - chose from method1 and method2 impact outcome can be provided as age specific if simplified=FALSE

`age_max` maximum age considered, eg. age_max = 4 for under 5s

modified_update_calculate

Do a modified update

Description

Do a modified update

Usage

```
modified_update_calculate(con, touchstone_name_mod, touchstone_use)
```

Arguments

con	Database connection
touchstone_name_mod	Name of the new modified touchstone that we are creating
touchstone_use	Name of the touchstone that we are basing this off of

modified_update_summary_output

Modified update summary output

Description

Modified update summary output

Usage

```
modified_update_summary_output(con, res, path_meta)
```

Arguments

con	Database connection
res	A modified update object (returned from `modified_update_calculate`)
path_meta	Path to the metadata directory. In this directory the following files must exist: `gavi_country_data.csv`, `tr_touchstone.csv`, and `years_output.csv`. There is no validation done on these files and providing the wrong thing will break in interesting and unknown ways.

mu_scale *Calculate updated impact*

Description

Calculate updated impact

Usage

```
mu_scale(name, d)
```

Arguments

name	Impact type: deaths_averted or cases_averted
d	Data: use impact_rate_tot (method 2)

mu_year_introduction *Find year of introduction*

Description

Look for introduction year and add to summary output

Usage

```
mu_year_introduction(con, dat, dat_summary)
```

Arguments

con	Database connection
dat	Data: the list output from the modup
dat_summary	Data: the summary output from the modup

prepare_recipe *Prepare impact recipe*

Description

Prepare impact calculation recipes

Usage

```
prepare_recipe(con, recipe = "impact.csv")
```

Arguments

`con`	Database connection. You will need to be `readonly` user to run this function.
`recipe`	at the moment, it is a csv file. Once imported, recipe will be more flexible

project_coverage *Project coverage*

Description

Project coverage

Usage

```
project_coverage(dat, year_project_from, year_from = 1980,
  year_to = 2100)
```

Arguments

`dat`	Data with columns...
`year_project_from`	This is the year that the projections start
`year_from`, `year_to`	Range of the actual coverage data that that you want.

Index