Package 'LTBIscreeningproject'

January 16, 2019

Type Package

Title LTBI screening cost-effectiveness analysis

Version 0.1.0.9000

Maintainer Nathan Green `<nathan.green@imperial.ac.uk>`

Description Decision tree and competing risks individual level model.

License MIT + file LICENSE

LazyData TRUE

Imports QALY,
   readr,
   ggplot2,
   readxl,
   data.tree,
   data.table,
   treeSimR,
   cmprsk,
   reshape,
   GGally,
   BCEA,
   purrr,
   plyr,
   dplyr,
   reshape2,
   magrittr,
   purrr,
   logging,
   arm,
   parallel,
   ggtern,
   miscUtilities,
   crayon,
   tibble,
   withr,
   assertr,
   devtools,
   lattice,
   here

RoxygenNote 6.1.1

Suggests knitr,
        rmarkdown,
        testthat,
        assertthat,
        denstrip,
        plotCostEffectiveness,
        memoise,
        directlabels,
        cowplot,
        gridExtra,
        covr,
        ggridges,
        lattice,
        grid

VignetteBuilder knitr

R **topics documented:**

activetb_qaly_cost   *activetb_qaly_cost*

Description

For the Population model, calculate various QALYs and costs accounting for active TB progression
in non-cured cohort subset.

Usage

```
activetb_qaly_cost(dectree_res, interv, cohort, folders = NA)
```

Arguments

`dectree_res`	Output of `parallel_decision_tree()`. This contains the probability of being cured of LTBI via screening.
`interv`	list of fixed model run parameter values
`cohort`	dataframe of individual level
`folders`	list of strings

Value

- QALY.statusquo: For each scenario a vector of total QALYs without screening programme, length number of sims. These are all the same because population QALYs are not varied for the cohort.

- QALY.screened: For each scenario a vector of total QALYs with screening programme, length number of sims.

- E_cost_screened: For each scenario single expected cost with screening programme.

- cost.screened_person: For each scenario a vector of QALYs per person with screening programme, length number of sims.

- cost.statusquo_person: For each scenario a vector of costs per person without screening programme, length number of sims.

- cost_incur: For each scenario a vector of incurred costs by screening programme, length number of sims.

- cost.statusquo: For each scenario a vector of total costs without screening programme, length number of sims. The are not identical because TB costs are randomly sampled.

- cost.screened: For each scenario a vector of total costs with screening programme, length number of sims.

- E_QALY_screened: For each scenario single expected QALYs with screening programme.

- QALY.screened_person: For each scenario a vector of QALYs per person with screening programme, length number of sims.

- QALY.statusquo_person: For each scenario a vector of QALYs per person without screening programme, length number of sims. These are all the same.

- QALYgain: For each scenario a vector of total QALYs gained with screening programme as the difference between screening and status-quo, length number of sims.

- cost_incur_person: For each scenario a vector of total costs incured per person with screening programme as the difference between screening and status-quo, length number of sims.

- E_cost_incur: For each scenario the expectedd total cost incured with screening programme as the difference between screening and status-quo.

- E_cost_incur_person: For each scenario the expected total cost incured per person with screening programme as the difference between screening and status-quo.

- QALYgain_person: For each scenario a vector of total QALY gained per person with screening programme as the difference between screening and status-quo.

- E_QALYgain: For each scenario the expected total QALY gained with screening programme as the difference between screening and status-quo.

- E_QALYgain_person: For each scenario the expected total QALy gained per person with screening programme as the difference between screening and status-quo.

append_scenario_num	*append_scenario_num*

Description

append_scenario_num

Usage

```
append_scenario_num(dat, i, excluded_cols = c("year", "discount"))
```

Arguments

excluded_cols

assign_LTBI_status	*Assign LTBI status from country of origin*

Description

Assign LTBI status from country of origin

Usage

```
assign_LTBI_status(IMPUTED_sample, pLatentTB.who)
```

Arguments

pLatentTB.who

base_filled_contour_grid	
	base_filled_contour_grid

Description

base_filled_contour_grid

Usage

```
base_filled_contour_grid(plot_data, folders)
```

Arguments

folders

bayeslm_wtp *Partial Bayesian linear regression function*

Description

Partial Bayesian linear regression function

Usage

```
bayeslm_wtp(nmb_mat, formula)
```

Arguments

...

bayes_predict *bayes_predict*

Description

bayes_predict

Usage

```
bayes_predict(model_fit, newdata, n_sim = 100)
```

Arguments

newdata

bcea_incremental *bcea_incremental*

Description

bcea_incremental

Usage

```
bcea_incremental(ce_incr)
```

Arguments

ce_incr

---

bcea_to_plotdata   *bcea_to_plotdata*

Description

to use in mash, contour plotting functions

Usage

```
bcea_to_plotdata(bcea, folders, wtp_threshold = "20000")
```

Arguments

bcea	BCEA package object
folders	list
wtp_threshold	default: 20000

---

boxplot_INMB   *boxplot_INMB*

Description

boxplot_INMB

Usage

```
boxplot_INMB(bcea, ...)

## S3 method for class 'bcea'
boxplot_INMB(bcea, folders = NA, wtp_threshold = 20000,
  oneway = FALSE)
```

Arguments

```
...
```

---

branch_unif_params   *branch_unif_params constructor*

Description

branch_unif_params constructor

Usage

```
branch_unif_params(pmin, pmax, name)
```

Arguments

name

---

calc.ICER   *Incremental Cost Effectiveness Ratio*

Description

Differences are 'intervention' - 'status-quo'.

Usage

```
calc.ICER(e, c, ref = 1)
```

Arguments

e	Effectiveness i.e. health e.g. QALYs
c	Costs
ref	Reference column. Defaults: 1

---

calc.INMB   *Incremental Net (Monetary) Benefit*

Description

Differences are 'intervention' - 'status-quo'.

Usage

```
calc.INMB(e, c, ref = 1, wtp = 20000)
```

Arguments

e	Effectiveness i.e. health e.g. QALYs
c	Costs
ref	Reference column. Default: 1
wtp	Willingness to pay threshold

Value

Vector

calc_num_active_tb   *Calculate Numbers of Active TB Cases using Hazards*

Description

Calculate the number tb_uk (extrapolated) using yearly hazards.

Usage

```
calc_num_active_tb(strat_pop, hazard)
```

Arguments

hazard

Details

##TODO: make dependent on different LTBI probs

calc_QALY_tb   *Calculate QALYs for active TB cases*

Description

Calculate the QALYs for each active TB individuals for each of 3 alternatives:

Usage

```
calc_QALY_tb(intervals = NA, utility, age, start_delay = NA,
  discount_rate = 0.035, ...)
```

Arguments

intervals	Time intervals for each utility
utility	(list) Utility value of non-diseased individual e.g. 1. Utility value of diseased individual
age	Ages in years
start_delay	What time delay to time origin, to shift discounting to smaller values
discount_rate	default 3.5%
...	Additional arguments

Details

- diseasefree: to all-cause death
- fatality: case-fatality 12 months from notification
- cured: successfully treated

Assume that death if it happens is within the first year of active TB. Assume that active TB cases when treated and survive first year are ~~fully cured~~.

Consider person-perspective (death) or NHS-perspective (exit uk) by defining the particular time-to-event end point.

Value

list of diseasefree, death, cured QALYs

---

care_cascade_prob *Plot care cascade*

Description

Plot care cascade

Usage

```
care_cascade_prob(...)

care_cascade_num(...)

plot_care_cascade(data_folder, plots_folder, prob_or_num, file_name)
```

Arguments

data_folder	string
plots_folder	string
prob_or_num	Probabilities or absolute numbers
file_name	

Examples

```
plot_care_cascade(parent_folder,
                  prob_or_num = "prob")
```

---

cbind_all_subsets *cbind_all_subsets*

Description

cbind_all_subsets

Usage

```
cbind_all_subsets(read_folder, write_folder = read_folder)
```

Arguments

read_folder	text string
write_folder	text string

---

cc_lot_kernel   *cc_lot_kernel*

Description

cc_lot_kernel

Usage

```
cc_lot_kernel(boxplot, dat)
```

Arguments

dat

---

ceac_plot_and_save   *Cost-effectiveness acceptability curve (CEAC) plot and save*

Description

Cost-effectiveness acceptability curve (CEAC) plot and save

Usage

```
ceac_plot_and_save(bcea, ...)

## S3 method for class 'bcea'
ceac_plot_and_save(bcea, folders, ...)
```

Arguments

...

---

ceplane_plot_and_save *ceplane_plot_and_save*

Description

ceplane_plot_and_save

Usage

```
ceplane_plot_and_save(bcea, ...)

## S3 method for class 'bcea'
ceplane_plot_and_save(bcea, folders, ...)
```

Arguments

...

ce_boundary_line_plot *ce_boundary_line_plot*

Description

ce_boundary_line_plot

Usage

```
ce_boundary_line_plot(plot_data, folders = NA,
  x_var = "Start_Treatment_p", y_var = "Complete_Treatment_p")
```

Arguments

plot_data

ce_boundary_points_plot

ce_boundary_points_plot

Description

ce_boundary_points_plot

Usage

```
ce_boundary_points_plot(plot_data, folders)
```

Arguments

folders

ce_default *ce_default*

Description

Uses the first column of the status-quo matrices for all status-quo comparisons.

Usage

```
ce_default(ce0, ce1)
```

Arguments

ce1

CE_plane_trajectories *CE_plane_trajectories*

Description

CE_plane_trajectories

Usage

```
CE_plane_trajectories(bcea, folders)
```

Arguments

folders

ce_plane_with_annotations

ce_plane_with_annotations

Description

ce_plane_with_annotations

Usage

```
ce_plane_with_annotations()
```

coefficient_plots *coefficient_plots*

Description

coefficient_plots

Usage

```
coefficient_plots(folders)
```

Arguments

folders

---

combine_cost_and_p_xlsheets

combine_cost_and_p_xlsheets

Description

combine_cost_and_p_xlsheets

Usage

```
combine_cost_and_p_xlsheets(parameter_p, parameter_cost)
```

Arguments

parameter_cost

---

combine_freq_tables   *combine_freq_tables*

Description

combine_freq_tables

Usage

```
combine_freq_tables(data_folder, file_name)
```

Arguments

file_name

Examples

```
combine_freq_tables(parent_folder,
                    file_name = "all_subsets.csv")

combine_freq_tables(parent_folder,
                    file_name = "prob_subset_dectree.csv")
```

---

combine_popmod_dectree_res

combine_popmod_dectree_res

Description

combine_popmod_dectree_res

Usage

```
combine_popmod_dectree_res(cohort, interv, popmod_res, dectree_res,
  folders = NA)
```

Arguments

folders

Value

- ce0: marginal status-quo

- ce1: marginal intervention

- ce_default: non_incremental cost-effectiveness

- ce_incr: incremental cost-effectivness

---

costeff_stats   *Cost-effectiveness Statistics*

Description

Cost-effectiveness Statistics

Usage

```
costeff_stats(scenario_dat, interv_QALY, interv_cost, pop_year)
```

Arguments

scenario_dat   list

pop_year

Value

list

---

cost_tb_notif *Total costs of first and secondary active TB cases*

Description

Total costs of first and secondary active TB cases

Usage

```
cost_tb_notif(num_sec_inf, unit_cost, notif_discounts,
  secondary_inf_discounts)
```

Arguments

num_sec_inf   Number of secondary infections for each index TB case; vector of integer (0 or
                  1)

unit_cost   Single value

notif_discounts
                  for each index TB case; vector

secondary_inf_discounts
                  for each index TB case; vector

---

count_comprsk_events *count_comprsk_events*

Description

Counts competing risk events over time. Replaces a (deprecated) previous rewritten version. Tied
times are prioritised according to their named order.

Usage

```
count_comprsk_events(event_times)
```

Arguments

event_times   List; tb, (fup), exit_uk, death

Value

data.frame

---

cp_in_data_to_out_dir   *Copy input data to output folder*

Description

Copy input data to output folder

Usage

```
cp_in_data_to_out_dir(file_names, to_dir)
```

Arguments

to_dir

---

create_and_save_policies

create_and_save_policies

Description

create_and_save_policies

Usage

```
create_and_save_policies(incidence_list, endpoints, LTBI_test, treatment)
```

Arguments

treatment

---

create_and_save_scenarios

Create and save scenarios

Description

Long flat array

Usage

```
create_and_save_scenarios(file_tag)
```

Arguments

file_tag   Trailing part of the Excel file name

Value

none (save to project data folder)

---

create_avoid_tb_list   *create_avoid_tb_list*

Description

Numbers of individuals who avoid getting TB due to screening

Usage

```
create_avoid_tb_list(scenario_res, n_all_tb, n_uk_tb)
```

Arguments

```
n_uk_tb
```

Value

list

---

create_pred_newdata   *Create prediction input data*

Description

This can be a super set of values because the prediction function just picks the ones that are in the fitted model. Although for a fine grid on some parameter this may result in a very large array.

Usage

```
create_pred_newdata(grid_min = NA, grid_max = NA, step_size = NA,
  agree = NA, sens = NA, spec = NA, start = NA, complete = NA,
  effective = NA, cost = NA)
```

Arguments

```
grid_min, grid_max
```
                   upper and lower limits
```
step_size
```
         equally spaced intervals
```
agree, start, complete, effective, cost
```
                   parameter names

Examples

```
create_pred_newdata(sens = 0.9,
                    spec = 0.85,
                    #start = c(0.5, 1),
                    #complete = c(0.5, 1),
                    cost = 50)
```

decision_tree_cluster   *Decision tree*

Description

Calculate decision tree expected costs and QALY loss for N simulations.

Usage

```
decision_tree_cluster(params, N.mc = 2,
  cost_dectree = "osNode_cost_2009.Rds",
  health_dectree = "osNode_health_2009.Rds", out_datatree = FALSE)
```

Arguments

params	long format array
N.mc	integer
cost_dectree	Rds file names (string)
health_dectree	Rds file names (string)
out_datatree	Output full datatree object? Default: FALSE

Value

list

dectree_to_dataframe   *dectree_to_dataframe*

Description

TODO: change data.tree:: to output _all_ nodes

Usage

```
dectree_to_dataframe(osNode)
```

Arguments

osNode

---

design_matrix   *design_matrix*

Description

Create a flat array from scenario inputs.

Usage

```
design_matrix(params)
```

Arguments

params

Value

dataframe

Examples

```
scenario_params_df %>%
 design_matrix()
```

---

diroutput   *form name of output folder*

Description

form name of output folder

Usage

```
diroutput(policy_name, interv)
```

Arguments

interv

expected_cost_QALY *expected_cost_QALY*

Description

expected_cost_QALY

Usage

```
expected_cost_QALY(cohort, means)
```

Arguments

means

freq_table_for_publication

freq_table_for_publication

Description

https://stackoverflow.com/questions/34587317/using-dplyr-to-create-summary-proportion-table-with-several- categorical-factor-v?utm_medium=organic&utm_source=google_rich_qa&utm_campaign=google_rich_qa

Usage

```
freq_table_for_publication(wide_df, row_vars, col_var)
```

Arguments

col_var

gg_care_cascade *gg_care_cascade*

Description

gg_care_cascade

Usage

```
gg_care_cascade(dat, plots_folder = NA, prob_or_num = "prob",
  policy_name = NA, grp = NA, box_plot = FALSE)
```

Arguments

box_plot

handle_try_error *handle_try_error*

Description

handle_try_error

Usage

```
handle_try_error(try_out)
```

Arguments

try_out

histogram_INMB *histogram_INMB*

Description

histogram_INMB

Usage

```
histogram_INMB(bcea, ...)

## S3 method for class 'bcea'
histogram_INMB(bcea, folders = NA,
  wtp_threshold = 20000)

## S3 method for class 'bcea'
ridgeslineplot_INMB(bcea, folders = NA,
  wtp_threshold = 20000)
```

Arguments

...

hist_progression_times

hist_progression_times

Description

hist_progression_times

Usage

```
hist_progression_times(dat)
```

Arguments

dat   Individual level cohort data

Examples

```
hist_progression_times(cohort)
```

IMPUTED_sample   *Cohort used in model after data cleaning and prep*

Description

Cohort used in model after data cleaning and prep

Usage

```
IMPUTED_sample
```

Format

A data frame with xxx rows and xxx variables:

rNotificationDate_issdt Time in days from entry to active TB for original data

notif_issdt.years Time in years from entry to active TB for original data

all_tb_issdt Simulated time in years from entry to active TB

notif_issdt.years Simulated time in years from entry to active TB in EWNI

exituk_tb.years Simulated time in years from entry to active TB after exit ...

---

inmb_from_bcea   *inmb_from_bcea*

Description

Using data from simulation (not regression predictions).

Usage

```
inmb_from_bcea(bcea, folders, wtp_threshold = c(10000, 20000, 30000))
```

Arguments

wtp_threshold

Value

list by wtp

---

inmb_levelplot   *inmb_levelplot*

Description

inmb_levelplot

Usage

```
inmb_levelplot(plot_data, formula = as.formula(INMB ~ Start_Treatment_p *
  Complete_Treatment_p), start = NA, complete = NA,
  levels_range = seq(-70, 50, by = 5), folders = NA,
  plot_type = "base")
```

Arguments

formula	Default: INMB ~ Start_Treatment_p*Complete_Treatment_p
plot_type	base or ggplot2

insert_dectree_cost *insert_dectree_cost*

Description

##TODO: use for prep script

Usage

```
insert_dectree_cost(names, costs, osNode)
```

Arguments

names   node labels

osNode

integerLHS *integerLHS*

Description

https://stat.ethz.ch/pipermail/r-help/2007-January/124143.html

Usage

```
integerLHS(n, intGroups)
```

Arguments

intGroups

Examples

```
integerLHS(10, list(1:10, 31:40))
integerLHS(5, list(1:10, 31:40))
integerLHS(2, list(1:10, 31:40))
integerLHS(5, list(1:20, 31:60, 101:115))
integerLHS(5, list(seq(2,20,2), 31:60, 101:115))
```

interv_constructor *interv_constructor*

Description

interv_constructor

Usage

```
interv_constructor(N.mc = 1, cluster = FALSE, use_discount = TRUE,
  no_students = FALSE, force_everyone_stays = FALSE,
  screen_with_delay = TRUE, MAX_SCREEN_DELAY = 5, FUP_MAX_YEAR = 100,
  screen_age_range = 18:35, year_cohort = "2009",
  incidence_grps_screen = c("(0,50]", "(50,150]", "(150,250]",
  "(250,350]", "(350,1e+05]"), min_screen_length_of_stay = 0,
  ENDPOINT_cost = "death", ENDPOINT_QALY = "death")
```

Arguments

N.mc	Global fixed constant; default 1
use_discount	Global fixed constant
screen_with_delay	
	Rather than screen _everyone_ on entry screen at random 0-5 years from entry
FUP_MAX_YEAR	Time horizon for active TB progression
year_cohort	year_cohort = '2012' is most recent complete year; largest cohort, corresponds with Pareek () LTBI risk
incidence_grps_screen	
	Modified in the deterministic sensitivity analysis but set default values
min_screen_length_of_stay	
	Modified in the deterministic sensitivity analysis but set default values
ENDPOINT_cost	Modified in the deterministic sensitivity analysis but set default values
ENDPOINT_QALY	Modified in the deterministic sensitivity analysis but set default values

is.death *Is Follow-up Time a Time of Death*

Description

Create event-type indicators

Usage

```
is.death(imputation_num, data, fup_limit = 19723)
```

Arguments

fup_limit	

Details

fup_limit <- 19723 is days from 1960-01-01 TODO is.exit_uk, is.fup_limit

leaf_df_by_name   *leaf_df_by_name*

Description

Subset to dataframe of terminal nodes by name.

Usage

```
leaf_df_by_name(osNode, node_name)
```

Arguments

node_name	Text string
dectree_df	my_ToDataFrameTypeCol() output

Value

dataframe of subset

list_to_BCEA   *list_to_BCEA*

Description

transform to BCEA package input format

Usage

```
list_to_BCEA(scenario_list, discount = 1)

list_to_BCEA_incr(scenario_list, discount = 1)
```

Arguments

discount

lm_list_to_df   *lm_list_to_df*

Description

Create wide output table using broom.

Usage

```
lm_list_to_df(fit)
```

Arguments

fit

---

lm_multi_wtp   *Linear multivariate regression varying willingness to pay*

Description

Linear multivariate regression varying willingness to pay

Usage

```
lm_multi_wtp(nmb_formula, nmb_mat, f_lm = lm, folders = NA)
```

Arguments

nmb_mat	list by wtp
f_lm	function type of regression; bayeslm_wtp, lm; default: lm
folders	

---

make_ce0   *make_ce0*

Description

make_ce0

Usage

```
make_ce0(popmod_res)
```

Arguments

popmod_res	

---

make_ce1   *make_ce1*

Description

make_ce1

Usage

```
make_ce1(popmod_res, t_dectree, sdiscount)
```

Arguments

sdiscount	

---

make_incremental_ce   *make_incremental_ce*

Description

make_incremental_ce

Usage

```
make_incremental_ce(popmod_res, t_dectree, sdiscount, folders = NA)
```

Arguments

folders

---

make_wide_INMB   *make wide INMB array from predictions*

Description

reshape

Usage

```
make_wide_INMB(pred, newdata)
```

Arguments

newdata

---

my_ceac_plot   *my_ceac_plot*

Description

my_ceac_plot

Usage

```
my_ceac_plot(he, comparison = NULL, pos = c(1, 0), graph = c("base",
  "ggplot2"))
```

Arguments

graph

my_ToDataFrameTable   *my_ToDataFrameTable*

Description

This is the same as the same named function in `data.tree` except it is not filtered by leaf.

Usage

```
my_ToDataFrameTable(x, ..., pruneFun = NULL)
```

Arguments

pruneFun

See Also

[ToDataFrameTable](#)

my_ToDataFrameTypeCol   *my_ToDataFrameTypeCol*

Description

This is the same as the same named function in `data.tree` except it is not filtered by leaf.

Usage

```
my_ToDataFrameTypeCol(x, ..., type = "level", prefix = type,
  pruneFun = NULL)
```

Arguments

pruneFun

See Also

[ToDataFrameTypeCol](#)

---

net_benefit_list   *Net benefit on c and e lists*

Description

Net benefit on c and e lists

Usage

```
net_benefit_list(e_list, c_list, wtp_threshold)
```

Arguments

wtp_threshold

---

nmb_contour_plot   *nmb_contour_plot*

Description

Single or multiple contour plot.

Usage

```
nmb_contour_plot(plot_data, folders, x_var = "Start_Treatment_p",
  y_var = "Complete_Treatment_p", facet_vars = c("Agree_to_Screen_p",
  "Effective_p"))
```

Arguments

folders	List
x_var	string
y_var	string
facet_vars	Vector of strings

---

nmb_matrix *Net monetary benefit matrix*

Description

Create input data for regressions.

Usage

```
nmb_matrix(ce1, ce0, folders = NA, design_mat = NA, wtp_min = 10000,
  wtp_max = 30000)
```

Arguments

ce1	interventions
ce0	status-quo
folders	list of strings
wtp_min, wtp_max	
	Willingness-to-pay limits; numeric

Details

This is _not_ incremental benefit.

Value

list of data.frames by wtp

---

nmb_multi_regn *nmb_multi_regn*

Description

Fit regression

Usage

```
nmb_multi_regn(nmb_mat, folders = NA, f_lm = lm, interactions = NA,
  centre_p = 90)
```

Arguments

nmb_mat	list by wtp
f_lm	lm or bayeslm_wtp; default: lm
centre	default: 90

---

nmb_predictions   *Net monetary benefit regression predictions*

Description

High-level create desing matrix, fit model & predict.

Usage

```
nmb_predictions(ce_res, folders, use_newdata = TRUE)
```

Arguments

ce_res	from combine_popmod_dectree_res()
use_newdata	default TRUE

Value

Array of regression predictions.

---

nmb_scenarios   *Net monetary benefit over scenarios*

Description

Create long array over multiple wtp.

Usage

```
nmb_scenarios(e0, c0, e1, c1, wtp)
```

Arguments

wtp

---

num_subset_dectree   *num_subset_dectree*

Description

num_subset_dectree

Usage

```
num_subset_dectree(cohort, dectree_res, diroutput = NA)
```

Arguments

diroutput

---

num_subset_tb *num_subset_tb*

Description

Uses dectree subset_pop output instead of separate montecarlo()

Usage

```
num_subset_tb(cohort, dectree_res, folder = NA)
```

Arguments

folder   name text string

---

num_subset_tb_wide *num_subset_tb_wide*

Description

Uses dectree subset_pop output instead of separate montecarlo()

Usage

```
num_subset_tb_wide(cohort, dectree_res, folder)
```

Arguments

folder   text string

Value

tibble

---

oneway_matrix *Generate One-Way Model Matrix*

Description

Otherwise known as one-factor-at-a-time. For full-factorial design use `expand.grid`.

Usage

```
oneway_matrix(mid, high, low)
```

Arguments

`mid`	Vector of middle values
`high`	Vector of high values
`low`	Vector of low values

Value

matrix

Examples

```
mid <- c(2,3,4,10)
high <- c(4,5,6,100)
low <- c(0,1,2,-2)

oneway_matrix(mid, high, low)
```

---

optimal_thresholds   *Find optimal thresholds*

Description

Find optimal thresholds

Usage

```
optimal_thresholds(lm_multi, covar, centre)
```

Arguments

centre

---

parallel_decision_tree

                          Parallel cost-effectiveness decision tree

---

Description

the output is also saved in Q:/R/cluster–LTBI-decision-tree because the alternative way of running
is on the DIDE cluster so all of the results are in the same place

Usage

```
parallel_decision_tree(scenario_params, interv, folders,
  out_datatree = FALSE)
```

Arguments

scenario_params
:   list of dataframes

interv
:   list of policy parameters

folders
:   list

out_datatree
:   default:FALSE

Details

Based on code here: https://www.r-bloggers.com/how-to-go-parallel-in-r-basics-tips/

Value

List of decision_tree_cluster outputs for each scenario

---

plots_and_tables_policies
plots_and_tables_policies

Description

plots_and_tables_policies

Usage

```
plots_and_tables_policies()
```

---

plots_and_tables_scenarios
plots_and_tables_scenarios

Description

Multiple types of figures and tables.

Usage

```
plots_and_tables_scenarios(cohort, dectree_res, popmod_res, ce_res,
  folders)
```

Arguments

folders
:   list of input/output folders

Value

Side effects only

---

plot_CE_contours   *plot_CE_contours*

Description

plot_CE_contours

Usage

```
plot_CE_contours(dat_INMB, folders)
```

Arguments

dat_INMB   from nmb_predictions()

folders   list

---

plot_QALY_cost_distns_by_scenario
                        plot_QALY_cost_distns_by_scenario

Description

QALY gains and cost incurred histograms for each scenario

Usage

```
plot_QALY_cost_distns_by_scenario(aTB_CE_stats, folders)
```

Arguments

folders

Examples

```
load("C:/Users/ngreen1/Dropbox/TB/LTBI/R/LTBIscreeningproject/ext-data/18_to_35_in_2009/programme_level_sce
load("C:/Users/ngreen1/Dropbox/TB/LTBI/R/LTBIscreeningproject/ext-data/18_to_35_in_2009/programme_level_sce
```

---

policy_cohort *Create policy cohort*

Description

Filter individuals by policy definition.

Usage

```
policy_cohort(cohort_in, policy_name, interv)
```

Arguments

cohort_in   total sample

interv   list of conditions

Value

cohort

---

policy_interv *policy_interv*

Description

Set the intervention parameter values within an evinonment.

Usage

```
policy_interv(policy_name, interv)
```

Arguments

policy_name   string

interv   list

---

predict_nmb_wtp *Predict from a list of fitted regressions*

Description

Predict from a list of fitted regressions

Usage

```
predict_nmb_wtp(fits_list, newdata = NA)
```

Arguments

newdata

prioritise_events	*prioritise_events*

Description

Replace same-year events with NA make sure that tb is always counted as priority

Usage

```
prioritise_events(times_dat)
```

Arguments

times_dat

prob_ce_gridplot	*prob_ce_gridplot*

Description

prob_ce_gridplot

Usage

```
prob_ce_gridplot(out_sim, formula = as.formula(prob_CE ~
  Start_Treatment_p * Complete_Treatment_p), folders = NA,
  plot_type = "base")
```

Arguments

formula	Defalut: prob_CE ~ Start_Treatment_p * Complete_Treatment_p
folders	list
plot_type	base or ggplot2

prob_from_cum_incidence

Calculate Jump Probabilities from Cumulative Incidence Functions

Description

see Jackson ()

Usage

```
prob_from_cum_incidence(cum_incidence_event, cum_incidence_comprisks)
```

Arguments

cum_incidence_event

                 Cumulative incidence for the event of interest

cum_incidence_comprisks

                 List of cumulative incidence for the other competing risk events

Value

Discrete probabilities

---

prob_subset_dectree   *prob_subset_dectree*

Description

prob_subset_dectree

Usage

```
prob_subset_dectree(cohort, dectree_res, diroutput = NA,
  LTBI_to_TB = 0.1)
```

Arguments

diroutput

---

remove_cols_constant_vars

                remove_cols_constant_vars

Description

remove_cols_constant_vars

Usage

```
remove_cols_constant_vars(nmb_mat)
```

Arguments

nmb_mat

---

ridgeslineplot_INMB   *ridges line plot INMB*

Description

ridges line plot INMB

Usage

```
ridgeslineplot_INMB(bcea, ...)
```

Arguments

...

---

rows_first_n_ids   *rows_first_n_ids*

Description

Finds the rows corresponding to the first n individuals by ascending id numbers.

Usage

```
rows_first_n_ids(id_avoid, prop_avoid)
```

Arguments

id_avoid   IDs, may have gaps/missing numbers

prop_avoid

Value

logical vector

---

run_final_message   *run_final_message*

Description

run_final_message

Usage

```
run_final_message(run)
```

Arguments

run

---

run_model   *Run model*

Description

Run model

Usage

```
run_model(cohort_data = NA, make_plots = TRUE, sink_out = FALSE)
```

Arguments

sink_out   output to file? Default: FALSE

---

run_policy   *run_policy*

Description

A single policy simulation

Usage

```
run_policy(cohort = NA, make_plots = TRUE)
```

Arguments

cohort_data

---

sample_subset_pop_dectree
                        sample_subset_pop_dectree

Description

Iteratively randomly samples probabilities and then calculates subset sizes

Usage

```
sample_subset_pop_dectree(osNode, n = 1, sample_p = TRUE)
```

Arguments

osNode	data.tree object
n	Sample size
sample_p	Random sample TRUE/FALSE

Value

matrix

---

sample_tb   *Sample (Updated) Active TB Status for Active TB Cases*

Description

Sample (Updated) Active TB Status for Active TB Cases

Usage

```
sample_tb(prob, is.tb = NA)
```

Arguments

prob   probability of success (e.g. completing treatment) for all cohort; i.e. probability
              of FALSE/0.

is.tb   is individual active TB case (logical)

Value

(counterfactual) TB status for active TB cases

---

sample_tb_year   *Sample active TB progression time*

Description

Given that an individual progresses then this approach samples active TB times.

Usage

```
sample_tb_year(fup_issdt, death_issdt, prob)
```

Arguments

fup_issdt   Time to follow-up/exit EWNI

death_issdt   Time to all-cause death (competing risk)

prob   Incidence density of progression

Details

Two-step mixture model for tb sampling:

1. Do they progress?
2. Sample TB time after follow-up

Value

Notification time

`save_session_info` *Save session info*

Description

Save session info

Usage

```
save_session_info(file)
```

Arguments

`file`	string

`scenario_cost` *Calculate total cost of a scenario*

Description

Calculate total cost of a scenario

Usage

```
scenario_cost(endpoint, unit_cost, probs_contact, cohort, prop_avoided)
```

Arguments

`endpoint`	`death` or `exit uk`
`unit_cost`	Diagnosis and treatment cost distributions
`probs_contact`	Proportions of individuals in subsets
`cohort`	nrow total number of tb cases in EWNI and after exit
`prop_avoided`	p_LTBI_to_cured

Value

list `statusquo`' and `screened`

---

scenario_QALY   *Calculate total QALYs of a scenario*

Description

Calculate total QALYs of a scenario

Usage

```
scenario_QALY(prop_avoided, endpoint, cohort, ordered = TRUE)
```

Arguments

prop_avoided	probability
endpoint	'death' or 'exit uk'
cohort	Individual data
ordered	Should individuals have a fixed order when avoiding tb; default: TRUE

Value

list of status-quo and screened life-time QALYs

---

scenario_QALYloss   *scenario_QALYloss*

Description

Splits output also into due to morbidity and mortality.

Usage

```
scenario_QALYloss(prop_avoided, endpoint, cohort)
```

Arguments

cohort

screened_cohort_cost   *Calculate Potentially Screened Cohort Costs*

Description

Substract the avoided cost of those successfully screened from status-quo cost.

Usage

```
screened_cohort_cost(n.diseasefree, cost.statusquo, unit_cost_case)
```

Arguments

n.diseasefree	Number of disease-free individuals
cost.statusquo	Cost under status-quo
unit_cost_case	Unit cost of detect and treat an active TB case

Value

Total cost for potentially screened cohort

See Also

screened_cohort_QALYs

---

screened_cohort_QALYs   *Calculate Potentially Screened Cohort QALYs*

Description

Calculate Potentially Screened Cohort QALYs

Usage

```
screened_cohort_QALYs(n.diseasefree, QALY)
```

Arguments

n.diseasefree	Number of disease-free individuals
QALY	List of QALYs for total cohort status-quo (assumed treated and cured), death, or all treated to disease-free

Value

Total QALYs for potentially screened cohort

See Also

screened_cohort_cost

screen_discount *screen_discount*

Description

Discount cost and QALYs in decision tree due to delayed start of screening from entry

Usage

```
screen_discount(cohort, discount_rate = 0.035)
```

Arguments

cohort   individual level data

discount_rate   default: 3.5%

setup_folders *Setup folders*

Description

Setup folders

Usage

```
setup_folders(policy_name, interv)
```

Arguments

policy_name   String

interv   List of model run constants

Value

List of folder locations

set_branch_uniform_params

set_branch_uniform_params

Description

set_branch_uniform_params

Usage

```
set_branch_uniform_params(vals, osNode)

## Default S3 method:
set_branch_uniform_params(vals, osNode)

## S3 method for class 'branch_unif_params'
set_branch_uniform_params(vals, osNode)

## S3 method for class 'test'
set_branch_uniform_params(vals, osNode)
```

Arguments

osNode

---

sim_tb_times   *Simulate TB progression times*

Description

Simulate TB progression times

Usage

```
sim_tb_times(data, prob)
```

Arguments

prob

split_time_intervals   *split_time_intervals*

Description

split_time_intervals

Usage

```
split_time_intervals(cohort, Tx_interval = 0.5)
```

Arguments

Tx_interval

stan_predict   *stan_predict*

Description

stan_predict

Usage

```
stan_predict(stan_fit, newdata, n_draws)
```

Arguments

n_draws

stan_predictions   *stan_predictions*

Description

stan_predictions

Usage

```
stan_predictions(ce_res, folders, use_newdata = TRUE)
```

Arguments

use_newdata

---

strat_pop_year *strat_pop_year*

Description

strat_pop_year

Usage

```
strat_pop_year(cohort, dectree_res, prop_avoid, folders)
```

Arguments

folders

---

subpop_by_exituk_year *subpop_by_exituk_year*

Description

Count number deaths & active TB cases in each exit uk year group

Usage

```
subpop_by_exituk_year(data)
```

Arguments

data   Individual-level sample?

---

subset_dectree *num_subset_dectree*

Description

Counts or proportion frequency of subset sizes along screening pathway.

Usage

```
subset_dectree(cohort, subset_pop, num_screen = 1)
```

Arguments

cohort   individual level data

subset_pop   part of output of decision_tree_cluster()

num_screen

Value

tibble

---

subset_pop_dectree   *Subset populations of decision tree*

Description

Specific to the LTBI screening model, this gives the total probabilities of particular state on the pathway by summing across nodes, using pathprobs.

Usage

```
subset_pop_dectree(osNode)
```

Arguments

osNode   data.tree object

Value

data.frame of probabilities

---

subset_pop_dectree_TEST

Subset populations of decision tree TEST

Description

THIS FUNCTION IS SLOW BUT CAN BE USED TO TEST AGAINST ALTERNATIVE VERSION subset_pop_dectree.

Usage

```
subset_pop_dectree_TEST(osNode)
```

Arguments

osNode   data.tree object

Details

Specific to the LTBI screening model, this gives the total probabilities of particular state on the pathway by summing across nodes, using pathprobs.

Value

data.frame of probabilities

table_costeffectiveness

Table of cost-effectiveness statistics

Description

Take BCEA package object as input and creates a policy summary table over scenarios.

Usage

```
table_costeffectiveness(bcea, ...)

## S3 method for class 'bcea'
table_costeffectiveness(bcea_out, wtp_threshold = 20000,
  ncohort = 1000, folder = NA)
```

Arguments

`bcea_out`	Pre-calculated output from BCEA package
`wtp_threshold`	Willingness to pay; Default: B#20,000
`ncohort`	hypothetical cohort size. This scales up small values to something more meaningful.
`folder`	text string save location

Value

data.frame with columns:

- Cost
- QALY
- Incremental_cost
- Incremental_QALY
- ICER
- INB
- ceac_WTP20k
- ceac_WTP25k
- ceac_WTP30k

Examples

```
ce_res <- combine_popmod_dectree_res(cohort, interv,
                                     popmod_res, dectree_res)
s_bcea <- screen_bcea(ce_res$ce_incr)
table_costeffectiveness(s_bcea)
```

---

table_tb_avoided   *table_tb_avoided*

Description

table_tb_avoided

Usage

```
table_tb_avoided(dectree_res, folder = NA)
```

Arguments

folder   text string

Value

EWNI and total 5

---

table_tb_avoided_wide *table_tb_avoided_wide*

Description

table_tb_avoided_wide

Usage

```
table_tb_avoided_wide(dectree_res, folder = NA)
```

Arguments

folder   text string

Value

EWNI and total 5

TB_burden_countries	*TB incidence in country from WHO 2017 report*

Description

2016 data

Usage

```
TB_burden_countries
```

Format

data.frame

Source

<http://www.who.int/tb/country/data/download/en/>

test	*test constructor*

Description

test constructor

Usage

```
test(sens, spec)
```

Arguments

spec

tornado_plot_ICER	*tornado_plot_ICER*

Description

tornado_plot_ICER

Usage

```
tornado_plot_ICER(bcea, ...)

## S3 method for class 'bcea'
tornado_plot_ICER(bcea, folders)
```

Arguments

...

tornado_plot_INMB   *tornado_plot_INMB*

Description

tornado_plot_INMB

Usage

```
tornado_plot_INMB(bcea, ...)

## S3 method for class 'bcea'
tornado_plot_INMB(bcea, folders)
```

Arguments

```
...
```

tornado_sim_plot   *tornado_sim_plot*

Description

tornado_sim_plot

Usage

```
tornado_sim_plot(ce_res, folders)
```

Arguments

```
folders          List
```

Value

ggplot object

WHO_incid_by_country_all_years

TB incidence in country from WHO all years up to 2016

Description

see also https://www.gov.uk/government/publications/tuberculosis-tb-by-country-rates-per-100000-pe
create_prev_inc_country_array.R

Usage

    WHO_incid_by_country_all_years

Format

data.frame

Source

http://apps.who.int/gho/data/node.main.1320?lang=en

Index