Proportional multi-state life table

-"Belen, Zapata-Diomedi belen.zapata-diomedi@rmit.edu.au" - "Alan, Both" -"Ali Abbas"

2020-11-26 07:23:44

```
##
## Attaching package: 'lubridate'
```

```
## The following objects are masked from 'package:base':
##
##     date, intersect, setdiff, union
```

```
## Loading required package: ggridges
```

```
## The ggjoy package has been deprecated. Please switch over to the
## ggridges package, which provides the same functionality. Porting
## guidelines can be found here:
## https://github.com/clauswilke/ggjoy/blob/master/README.md
```

```
##
## Attaching package: 'ggpubr'
```

```
## The following object is masked from 'package:plyr':
##
##     mutate
```

```
## Loading required package: xts
```

```
## Loading required package: zoo
```

```
##
## Attaching package: 'zoo'
```

```
## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric
```

```
##
## Attaching package: 'PerformanceAnalytics'
```

```
## The following object is masked from 'package:graphics':
##
##     legend
```

```
##
## Attaching package: 'dplyr'

## The following objects are masked from 'package:xts':
##
##     first, last

## The following objects are masked from 'package:plm':
##
##     between, lag, lead

## The following objects are masked from 'package:plyr':
##
##     arrange, count, desc, failwith, id, mutate, rename, summarise,
##     summarize

## The following object is masked from 'package:kableExtra':
##
##     group_rows

## The following objects are masked from 'package:stats':
##
##     filter, lag

## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union

## Loading required package: lattice

## Loading required package: survival

## Loading required package: Formula

##
## Attaching package: 'Hmisc'

## The following objects are masked from 'package:dplyr':
##
##     src, summarize

## The following objects are masked from 'package:plyr':
##
##     is.discrete, summarize

## The following objects are masked from 'package:base':
##
##     format.pval, units
```

```
rm (list = ls())

mypath <- 'C:/Metahit/'
relative_path_execute <- paste0(mypath,'mh-execute')
relative_path_gbd <- paste0(mypath,'mh-mslt/input/gbd/GBD2017/METAHIT/')
relative_path_mslt <- paste0(mypath,'mh-mslt/')
```

```
## Global options
options(scipen=999)
options("getSymbols.warning4.0"=FALSE)
options("getSymbols.yahoo.warning"=FALSE)
knitr::opts_chunk$set(echo = TRUE, width = 100)
knitr::opts_chunk$set(message = FALSE)
knitr::opts_chunk$set(warning = FALSE)
knitr::opts_chunk$set(eval = TRUE)
knitr::opts_chunk$set(error = TRUE)
```

Introduction

This documnets consists of the following sections: 1) Generation inputs for the proportional multi-state life table (PMSLT); 2) Code explained to run the PMSLT for one area (e.g. city regions, country); 3) Final function to use in mh-execute to run the PMSLT; 4) Examles of output graphs

Generation inputs for the proportional multi-state life table (PMSLT)

Data inputs for the proportional multi-state life table model (PMSLT) include inputs for: general life table (population, mortality rates and prevalent years lived with disability rates for all causes (YLD)), diseases life tables (incidence and case fatality) and injuries process (mortality rates and years lived with disability rates). Table 1 describes all data inputs for the MSLT. Each of the MSLT components and related code are explained below Table 1. Data inputs were generated for: United Kingdon, England, Wales, Scotland, England Regions () and English city regions/Combined Authorities (). We used Global Burden of Disease data [@RN1] for all data inputs, and an Disbayes to derive case fatality (not available in GBD). GBD data is available for rates per 100,000 people, total numbers and percentage by age groups (5-years smallest category) and sex. Percentage represents proportion of the contribution of a diseases to the total burden, we do not use percentages.

Table 1: Table 1: PMSLT inputs

Component	Data needs	Input data	Data processing

General life table	Population numbers and mortality rates	Population and all-cause mortality by age-group (5 years) and sex	Population numbers by 5-year age groups and sex were derived from GBD data for rates (per 100,000) and numbers data (i.e. number*100,000/rate). For the English city regions data is avaialbe for Lower Tier Local Authority (LTLA) and Upper Tier Local Authority (UTLA). Population numbers for LTLA and UTLA were aggregated to City Regions. For UK, Scotland, Wales and England Regions no futher steps were required. Mortality rates per one were derived by diving all-cause mortality rates by population numbers by age and sex. Interpolation was used to derived single-year mortality rates from 5-year data.
General life table	All-cause YLDs	All-cause pYLDs by 5-year age groups and sex	Rates per one derived from numbers and population data and interpolated to one-year age groups. Same aggreagation process as for mortlaity was used for City Regions.
Disease life table	Diseases: Disability weights, incidence and case fatality	Disease specific pYLDs and prevalence. Incidence and case fatality were derived using Disbayes	Section XX outlines Disbayes process. Disability weights were calculated by dividing disease specific pYLDs by prevalence and adjusted by all cause pYLDs for 5-year age groups and sex and then interpolated to 1-year age groups.
Injuries: pYLDs and mortality rates	PYLDs and mortality rates for road injures for pedestrians, cyclist, COMPLETE	pYLD and mortality numbers by 5-year age group and sex	Rates per one derived from numbers and population data and interpolated to 1-year age groups
Diseases trends	TO DO	TO DO	Explained in sections Trends

Table 2 is an overview of inputs, ouputs and code for the generation of data inputs.

Table 2: Tabl

R file	Function	Input variable	Input file
mh-mslt\R\prepare_GEO.R	GetGEO	data_local	mh-execute\input\mh_regi
mh-mslt\R\organise_GBD.R	GetGBD	path and local_goverment_area	mh-mslt\input\gbd\META
mh-mslt\R\get_disease_names.R	GetDiseaseTable	disease_names_execute	mh-execute\inputs\dose_re
mh-metahit\R\prepare_GBD.R	calculateGBDwider	gbd, local and government_areas	gbd and local_goverment_
mh-metahit\R\prepare_GBD.R	calculateMSLT	gbd_wider and location-disease	gbd_wider, location (cityre

The below code generates inputs for: United Kingdom, England, Wales, Northen Ireland, Scotland, Wales, English regions (East Midlands, East of England, Greater London, North East England, North West England, South East England, South West England, West Midlands, Yorkshire and the Humber) and the English City Regions (Liverpool, Nottingham, Bristol, Northeast, Greater Manchester, Sheffield, West Midlands, Leeds and London). Raw inputs are in "inputs" folder. Processed data is in "output" folder. Final data save in mh-execute inputs/mslt. *GBD will update data end of October*

Read look-up table mapping between local authorities and city regions, and add names for regions that are not city regions. For the city regions we add up data available for lower and upper tier local authorithy areas.

```
source(paste0(relative_path_mslt, "R/prepare_GEO.R"))
data_local <- read_csv(file.path(relative_path_execute, "inputs/mh_regions_lad_lookup.csv"))
local_government_areas <- GetGEO(data_local)
head(local_government_areas)
```

Read original GBD data for 2017 and map to local goverment area. This step is requiered to map local goverment areas to gbd data and then to city regions. UPDATE with new release Oct 2020. DISBAYES INPUTS WILL ALSO NEED UPDATING. DISCUSS WITH TEAM.

```
source(paste0(relative_path_mslt, "R/organise_GBD.R"))
gbd <- getGBD(relative_path_gbd, local_government_areas)
head(gbd)
```

Read disease coding table.

```
source(paste0(relative_path_mslt, "R/get_disease_names.R"))
disease_names_execute=read_csv(file.path(relative_path_execute, "inputs/dose_response/disease_outcomes_
DISEASE_SHORT_NAMES <- GetDiseaseTable(disease_names_execute)

head(DISEASE_SHORT_NAMES)

saveRDS(DISEASE_SHORT_NAMES, paste0(relative_path_mslt, "output/parameters/DISEASE_SHORT_NAMES.rds"))
```

Data processing

Steps and code to generate inputs to run PMSLT from above generated files: gbd, local_goverment_areas and DISEASE_SHORT_NAMES.

Calculate baseline data by area

Orginal GBD data tidy up for later processing for PMSLT dataframe and disbayes inputs. Generates dataframe for all areas (city regions, UK, countries and English Regions). Inputs are by age (5-year groups) and sex.

```
source(paste0(relative_path_mslt, "R/prepare_GBD.R"))

gbd_wider <- calculateGBDwider(gbd = gbd, local_government_areas = local_government_areas)
head(gbd_wider)
```

Calculate baseline data for PMSLT

A PMSLT dataframe of inputs is generated from above dataframe by single year of age and adding Disbayes outputs. Disbayes output generated by Chris J. Data saved in input folder of mh-execute.

```
source(paste0(relative_path_mslt, "R/prepare_GBD.R"))

### Please check commment manual entry for function calculateMSLT
### Load disbayes data
load("C:/Metahit/mh-mslt/input/city regions/Output disbayes/cityregions_smoothed_res.rda")

mslt_inputs <- list()
index <- 1
for (loc in unique(local_government_areas$cityregion)) {

mslt_inputs[[index]] <- calculateMSLT(gbd_wider, location = loc , disease)

names(mslt_inputs)[index] <- paste0(loc)

write.csv(mslt_inputs[[index]], file=paste0(relative_path_execute, "/inputs/mslt/", loc, "_mslt",".csv"


index <- index + 1

}
```

Code explained to run the PMSLT for one area (e.g. city regions, country);

Inputs

1) mslt_df: above generated.
2) model parameters: age and sex cohort.
3) pifs: generated in mh-execute
4) relative risks diabetes: from literature

Define model parameters

```
### Model paramters
i_age_cohort <-  c(17, 22, 27, 32, 37, 42, 47, 52, 57, 62, 67, 72, 77, 82, 87, 92, 97)
i_sex <- c('male', 'female')
```

Get potential impact fractions from mh-execute

Pifs are used in the sceanrio and non-disease (road injuries and lower respiratory disease) life tables to modify incidence of diseases and mortality and yld rates for non-diseases. For now this is place holder. We can generate pifs as an outcome of mh-execute and save in mslt-repo to run independently for each of the city regions.

```
pif_expanded <- read_csv(paste0(relative_path_mslt, "/input/pif.csv")) %>%
  mutate(age_cat = case_when(age_cat =="16-19" ~ 17, ### Create matching age cohort variable to match m
                        age_cat =="20-24" ~ 22,
                        age_cat =="25-29" ~ 27,
                        age_cat =="30-34" ~ 32,
                        age_cat =="35-39" ~ 37,
                        age_cat =="40-44" ~ 42,
                        age_cat =="45-49" ~ 47,
                        age_cat =="50-54" ~ 52,
                        age_cat =="55-59" ~ 57,
                        age_cat =="60-64" ~ 62,
                        age_cat =="65-69" ~ 67,
                        age_cat =="70-74" ~ 72,
                        age_cat =="75-79" ~ 77,
                        age_cat =="80-84" ~ 82,
                        age_cat =="85-89" ~ 87,
                        age_cat =="90-94" ~ 92,
                        age_cat =="95-120" ~ 97)) %>%
                  dplyr::rename(pif_ihd=scen_pif_pa_ap_noise_no2_ihd, ### rename all original pif colum
                  pif_stroke=scen_pif_pa_ap_stroke,
                  pif_colon=scen_pif_pa_colon,
                  pif_t2d=scen_pif_pa_t2d,
                  pif_endo=scen_pif_pa_endo,
                  pif_lc=scen_pif_pa_ap_lc,
                  pif_lri_deaths=scen_pif_ap_lri,
                  pif_copd=scen_pif_ap_copd,
                  pif_breast=scen_pif_pa_breast,
                  pif_cyclist_deaths=scen_cyclist_Fatal,
                  pif_pedestrian_deaths=scen_pedestrian_Fatal,
                  pif_cyclist_ylds=scen_cyclist_Serious,
                  pif_pedestrian_ylds=scen_pedestrian_Serious,
                  pif_motor_deaths='scen_car/taxi_Fatal',
                  pif_motorcyclist_deaths='scen_motorcycle_Fatal',
                  pif_motor_ylds='scen_car/taxi_Serious',
                  pif_motorcyclist_ylds=scen_motorcycle_Serious) %>%
                  mutate(pif_lri_ylds=pif_lri_deaths) %>% ###Repeat pif lri for deaths and ylds
                  dplyr::slice(rep(1:dplyr::n(), each = 5)) %>% ### expand to 1-yr group (repeat values
                  mutate(age=rep(seq(16,100,1), times = 2))
```

Get relative risks for diabetes on cardiovascular diseases

Diabetes is a risk factor for cardiovascular diseases. Parametric and probabilistic inputs. To be included in metahit_functions.R Variables with normal distribution.

```
### Parametric input
    DIABETES_IHD_RR_F <- 2.82 ## c(2.82, CI (2.35, 3.38) get SD from CI
    DIABETES_STROKE_RR_F <- 2.28 ## c(2.28) CI (1.93, 2.69) get SD from CI
    DIABETES_IHD_RR_M <- 2.16 ## c(2.16, CI (1.82, 2.56) get SD from CI
    DIABETES_STROKE_RR_M <- 1.83 ## c(1.83) CI (1.60, 2.08) get SD from CI

### Probabilitic input
# DIABETES_IHD_RR_F <- c(2.82, GetStDevRR(2.82, 2.35, 3.38))
# DIABETES_STROKE_RR_F <- c(2.28, GetStDevRR(2.28, 1.93, 2.69))
# DIABETES_IHD_RR_M <- c(2.16, GetStDevRR(2.16, 1.82, 2.56))
# DIABETES_STROKE_RR_M <- c(1.83, GetStDevRR(1.83, 1.60, 2.08))
```

Run general life table

Inputs for a general life table are population numbers and mortality rates. The below process life tables with life year, health-adjusted life years, life expectancy and health-adjusted life expectancy for each age and sex cohort for each city region. This code can be used independently if interested in life table analysis only. Select an area of interest. Options are: liverpool, nottingham, bristol, northeast, greatermanchester, sheffield westmidlands, leeds, london, United Kingdom, England, East Midlands, East of England, Greater London, North East England, North West England, South East England, South West England , West Midlands, Yorkshire and the Humber, Northern Ireland, Scotland, Wales

```
source(paste0(relative_path_mslt, "R/functions_MSLT.R"))

# dataframe of the age and sex cohorts (crossing just does a cross product) for loop below
  age_sex_cohorts <- crossing(data.frame(age=i_age_cohort),
                              data.frame(sex=c('male', 'female'))) %>%
    dplyr::mutate(cohort=paste0(age,"_",sex))

### Select inputs for area of interest

mslt_df <- mslt_inputs[["bristol"]]

 general_life_table_list_bl <- list()


  for (i in 1:nrow(age_sex_cohorts)){
    suppressWarnings(
      general_life_table_list_bl[[i]] <- RunLifeTable(
        in_idata    = mslt_df,
        in_sex      = age_sex_cohorts$sex[i],
        in_mid_age  = age_sex_cohorts$age[i],
        death_rates = NA ## mortality trends data if available, not for now. We would need trends for e
      ))
    names(general_life_table_list_bl)[i] <- age_sex_cohorts$cohort[i]
  }

  # convert the list of dataframes to single dataframes
  general_life_table_bl <- bind_rows(general_life_table_list_bl, .id = "age_group") %>%
    mutate(age_group = as.numeric(gsub("_.*","",age_group)))
```

Run baseline disease life tables

```
source(paste0(relative_path_mslt, "R/functions_MSLT.R"))


  disease_cohorts <- DISEASE_SHORT_NAMES %>%
    # Exclude non-diseases, road injuries, and diseases with no pif
    dplyr::filter(is_not_dis == 0 & acronym != 'no_pif' & acronym != 'other' ) %>%
    dplyr::select(sname,acronym,males,females)

  # adding the age and sex cohorts:
  age_sex_disease_cohorts <- crossing(age_sex_cohorts,disease_cohorts) %>%
    mutate(cohort=paste0(age,'_',sex,'_',sname)) %>%
    # Exclude non-male diseases (and non-female if there were any)
    filter( (sex=='male' & males==1) | (sex=='female' & females==1)) %>%
    dplyr::select(age,sex,sname,acronym,cohort) %>%
    # ishd and strk have the prerequisite disease dmt2
    mutate(prerequsite=ifelse(sname %in% c("ishd","strk"),paste0(age,"_",sex,"_dmt2"),0)) %>%
    # ensuring prequisites are calculated first
    arrange(age,sex,prerequsite,sname)


  disease_life_table_list_bl <- list()

  for (i in 1:nrow(age_sex_disease_cohorts)){
    disease_life_table_list_bl[[i]] <- RunDisease(
      in_idata         = mslt_df,
      in_mid_age       = age_sex_disease_cohorts$age[i],
      in_sex           = age_sex_disease_cohorts$sex[i],
      in_disease       = age_sex_disease_cohorts$sname[i],
      incidence_trends = NA,
      mortality_trends = NA
    )
    names(disease_life_table_list_bl)[i] <- age_sex_disease_cohorts$cohort[i]
  }
```

Run baseline non-disease life tables

```
non_disease_cohorts <- DISEASE_SHORT_NAMES %>%
  # Exclude non-diseases, road injuries, and diseases with no pif
  dplyr::filter(is_not_dis == 1 & acronym != 'no_pif' & acronym != 'other' ) %>%
  dplyr::select(sname,acronym,males,females)

# adding the age and sex cohorts:
age_sex_non_disease_cohorts <- crossing(age_sex_cohorts,non_disease_cohorts) %>%
  mutate(cohort=paste0(age,'_',sex,'_',sname)) %>%
  dplyr::select(age,sex,sname,acronym,cohort)

non_disease_life_table_list_bl <- list()

for (i in 1:nrow(age_sex_non_disease_cohorts )){

  non_disease_life_table_list_bl[[i]] <- RunNonDisease(
    in_idata         = mslt_df,
    in_sex           = age_sex_non_disease_cohorts$sex[i],
    in_mid_age       = age_sex_non_disease_cohorts$age[i],
    in_non_disease   = age_sex_non_disease_cohorts$sname[i]
  )
  names(non_disease_life_table_list_bl)[i] <- age_sex_non_disease_cohorts$cohort[i]
}
```

Run scenarios diseases life tables

Re run diseases life tables with modified incidence rates by the pif. New incidence rates = bl incidence rates * (1-disease_pif).

```
source(paste0(relative_path_mslt, "R/functions_MSLT.R"))


  disease_relative_risks <- tribble(
    ~sex    , ~prerequsite, ~disease , ~relative_risk       ,
    "male"  ,  "dmt2"     ,  "ishd"   ,  DIABETES_IHD_RR_M   ,
    "female",  "dmt2"     ,  "ishd"   ,  DIABETES_IHD_RR_F   ,
    "male"  ,  "dmt2"     ,  "strk"   ,  DIABETES_STROKE_RR_M,
    "female",  "dmt2"     ,  "strk"   ,  DIABETES_STROKE_RR_F
  )

  disease_life_table_list_sc <- list()

  for (i in 1:nrow(age_sex_disease_cohorts)){
    # i=6
    td1_age_sex <- mslt_df %>% ### new mslt dataframe with modified incidence rates
      filter(age >= age_sex_disease_cohorts$age[i] & sex == age_sex_disease_cohorts$sex[i])

    pif_colname <- paste0('pif_',age_sex_disease_cohorts$acronym[i])

    pif_disease <- pif_expanded %>%
      filter(age >= age_sex_disease_cohorts$age[i] & sex == age_sex_disease_cohorts$sex[i]) %>%
      dplyr::select(age,sex,pif_colname)

    # adjustment for diabetes effect on ihd and stroke
    if(age_sex_disease_cohorts$prerequsite[i] != 0){
      # get name for pif column
      target_disease <- paste0("pif_",age_sex_disease_cohorts$acronym[i])
      # get prerequisite disease cohort name (i.e., age_sex_dmt2 for diabetes)
      dia_col <- age_sex_disease_cohorts$prerequsite[i]
      # select relative risk of disease given diabetes (depends on sex, not age)
      relative_risk <- disease_relative_risks %>%
        filter(sex == age_sex_disease_cohorts$sex[i] &
                 disease == age_sex_disease_cohorts$sname[i]) %>%
        pull(relative_risk)
      # (store old pif)
      # old_pif <- pif_disease[[target_disease]]
      # diabetes pif = - { scenario prevalence - baseline prevalence } * (RR - 1)  / { baseline prevale
      scenario_prevalence <- disease_life_table_list_sc[[dia_col]]$px
      baseline_prevalence <- disease_life_table_list_bl[[dia_col]]$px
      pif_dia <- -(scenario_prevalence - baseline_prevalence)*(relative_risk-1)/
        (baseline_prevalence * (relative_risk-1) + 1)
      # modify pif for target disease: new pif =  (1 - old pif) * (1 - diabetes pif)
      pif_disease[[target_disease]] <- 1- (1-pif_disease[[target_disease]]) * (1-pif_dia)
      # print(sum(old_pif-pif_disease[[target_disease]]))
    }

    incidence_colname <- paste0('incidence_', age_sex_disease_cohorts$sname[i])
    new_col <- td1_age_sex%>%pull(incidence_colname) * (1 - (pif_disease%>%pull(pif_colname)))
    new_col[is.na(new_col)] <- 0
    td1_age_sex[[incidence_colname]] <- new_col

    ## Instead of idata, feed td to run scenarios. Now all diseases are run again, with the effect of d
    ## on cardiovascular diseases taken into account.

    disease_life_table_list_sc[[i]] <- RunDisease(
      in_idata         = td1_age_sex,
      in_sex           = age_sex_disease_cohorts$sex[i],
      in_mid_age       = age_sex_disease_cohorts$age[i],
      in_disease       = age_sex_disease_cohorts$sname[i],
      incidence_trends = NA,
      mortality_trends = NA
    )
    names(disease_life_table_list_sc)[i] <- age_sex_disease_cohorts$cohort[i]
  }



  for (cohort in age_sex_disease_cohorts$cohort) {
    disease_life_table_list_sc[[cohort]]$diff_inc_disease <-
      disease_life_table_list_sc[[cohort]]$incidence_disease - disease_life_table_list_bl[[cohort]]$inci

    disease_life_table_list_sc[[cohort]]$diff_prev_disease <-
      disease_life_table_list_sc[[cohort]]$px - disease_life_table_list_bl[[cohort]]$px

    disease_life_table_list_sc[[cohort]]$diff_mort_disease <-
      disease_life_table_list_sc[[cohort]]$mx - disease_life_table_list_bl[[cohort]]$mx

    disease_life_table_list_sc[[cohort]]$diff_pylds_disease <-
      (disease_life_table_list_sc[[cohort]]$px - disease_life_table_list_bl[[cohort]]$px) *
      (disease_life_table_list_bl[[cohort]]$dw_disease)
  }


  # convert the list of dataframes to single dataframes
  disease_life_table_bl <- bind_rows(disease_life_table_list_bl, .id = "age_sex_disease_cohort") %>%
    mutate(age_sex_disease_cohort = as.numeric(gsub("_.*","",age_sex_disease_cohort))) %>%
    dplyr::rename(age_group=age_sex_disease_cohort,
           cause=disease)

  disease_life_table_sc <- bind_rows(disease_life_table_list_sc, .id = "age_sex_disease_cohort") %>%
    mutate(age_sex_disease_cohort = as.numeric(gsub("_.*","",age_sex_disease_cohort))) %>%
   dplyr::rename(age_group=age_sex_disease_cohort,
           cause=disease)
```

Run scenarios non-diseases life tables

Re run non_diseases life tables with modified mortality and ylds rates by the pif. New rates = bl rates * (1-non_disease_pif). TO DO: CHECK CALCULATIONS OF NON DISEASE PIFS. CALCULATION SCENARIO_NUMBERS/BASELINE_NUMBER?

```
  non_disease_life_table_list_sc <- list()

 for (i in 1:nrow(age_sex_non_disease_cohorts)){
  # i=6
  td1_age_sex <- mslt_df %>% ### new mslt dataframe with modified injuries and lri deaths and ylds ra
    filter(age >= age_sex_non_disease_cohorts$age[i] & sex == age_sex_non_disease_cohorts$sex[i])

  pif_colname_deaths <- paste0('pif_',age_sex_non_disease_cohorts$acronym[i], '_deaths')
  pif_colname_ylds <- paste0('pif_',age_sex_non_disease_cohorts$acronym[i], '_ylds')

  pif_non_disease <- pif_expanded %>%
    filter(age >= age_sex_non_disease_cohorts$age[i] & sex == age_sex_non_disease_cohorts$sex[i]) %>%
    dplyr::select(age,sex,pif_colname_deaths,pif_colname_ylds)


  death_colname <- paste0('deaths_rate_', age_sex_non_disease_cohorts$sname[i])
  ylds_colname <- paste0('ylds_rate_', age_sex_non_disease_cohorts$sname[i])


  new_deaths <- td1_age_sex%>%pull(death_colname) * (1 - (pif_non_disease%>%pull(pif_colname_deaths))
  new_deaths[is.na(new_deaths)] <- 0

  new_ylds <- td1_age_sex%>%pull(ylds_colname) * (1 - (pif_non_disease%>%pull(pif_colname_ylds)))
  new_ylds[is.na(new_ylds)] <- 0

  td1_age_sex[[death_colname]] <- new_deaths
  td1_age_sex[[ylds_colname]] <- new_ylds

  ## Instead of idata, feed td to run scenarios. Now all non_diseases are run again


  non_disease_life_table_list_sc[[i]] <- RunNonDisease(
    in_idata         = td1_age_sex,
    in_sex           = age_sex_non_disease_cohorts$sex[i],
    in_mid_age       = age_sex_non_disease_cohorts$age[i],
    in_non_disease   = age_sex_non_disease_cohorts$sname[i]
  )
  names(non_disease_life_table_list_sc)[i] <- age_sex_non_disease_cohorts$cohort[i]

}

### Difference rates

 for (cohort in age_sex_non_disease_cohorts$cohort) {
   non_disease_life_table_list_sc[[cohort]]$diff_mort <-
     non_disease_life_table_list_sc[[cohort]]$deaths_rate -  non_disease_life_table_list_bl[[cohort]]$

   non_disease_life_table_list_sc[[cohort]]$diff_pylds <-
      non_disease_life_table_list_sc[[cohort]]$ylds_rate -  non_disease_life_table_list_bl[[cohort]]$



# convert the list of dataframes to single dataframes
```

```
  non_disease_life_table_bl <- bind_rows(non_disease_life_table_list_bl, .id = "age_sex_non_disease_coho
    mutate(age_sex_non_disease_cohort = as.numeric(gsub("_.*","",age_sex_non_disease_cohort))) %>%
    dplyr::rename(age_group=age_sex_non_disease_cohort,
           cause=non_disease,
           mx=deaths_rate)

  non_disease_life_table_sc <- bind_rows(non_disease_life_table_list_sc, .id = "age_sex_non_disease_coho
    mutate(age_sex_non_disease_cohort = as.numeric(gsub("_.*","",age_sex_non_disease_cohort))) %>%
    dplyr::rename(age_group=age_sex_non_disease_cohort,
           cause=non_disease,
           mx=deaths_rate)
```

Collect changes in mortality and prevalent years lived with disability.

```
### Diseases: Sum mortality rate and pylds change scenarios
  mx_pylds_sc_total_disease_df <- disease_life_table_sc %>%
    group_by(age_group,sex,age) %>%
    dplyr::summarise(mortality_sum=sum(diff_mort_disease,na.rm=T),
              pylds_sum=sum(diff_pylds_disease,na.rm=T)) %>%
    ungroup() %>%
    mutate(age_sex_cohort=paste0(age_group,'_',sex))

### Non-diseases
  mx_pylds_sc_total_non_disease_df <- non_disease_life_table_sc %>%
    group_by(age_group,sex,age) %>%
    dplyr::summarise(mortality_sum=sum(diff_mort,na.rm=T),
              pylds_sum=sum(diff_pylds,na.rm=T)) %>%
    ungroup() %>%
    mutate(age_sex_cohort=paste0(age_group,'_',sex))
```

Run scenario life table

General life table re-runed with modified mortality and Pylds rates

```
  general_life_table_list_sc <- list()

  for (i in 1:nrow(age_sex_cohorts)){
    # modify idata's mortality and pyld total for the said scenario
    mx_pylds_sc_total_disease_df_cohort <- mx_pylds_sc_total_disease_df %>%
      filter(age_sex_cohort==age_sex_cohorts$cohort[i]) %>%
      dplyr::select(age,mortality_sum,pylds_sum)

       mx_pylds_sc_total_non_disease_df_cohort <- mx_pylds_sc_total_non_disease_df %>%
      filter(age_sex_cohort==age_sex_cohorts$cohort[i]) %>%
      dplyr::select(age,mortality_sum,pylds_sum)

    ### Modify rates in static MSLT  (pylds are always static, mx can include future trends)
    #### With diseases changes in mortality and ylds
    td2 <- mslt_df %>%
      filter(sex==age_sex_cohorts$sex[i]) %>%
      left_join(mx_pylds_sc_total_disease_df_cohort,by="age") %>%
      mutate(mx=mx+replace_na(mortality_sum,0),
             pyld_rate=pyld_rate+replace_na(pylds_sum,0)) %>%
      dplyr::select(-mortality_sum,-pylds_sum)

    #### With diseases changes in mortality and ylds
    td3 <-  td2 %>%
      filter(sex==age_sex_cohorts$sex[i]) %>%
      left_join(mx_pylds_sc_total_non_disease_df_cohort,by="age") %>%
      mutate(mx=mx+replace_na(mortality_sum,0),
             pyld_rate=pyld_rate+replace_na(pylds_sum,0)) %>%
      dplyr::select(-mortality_sum,-pylds_sum)

    # ### Modify death rates with future trends NOT AVAILABLE FOR METAHIT
    # td3 <- death_projections %>%
    #   mutate(cohort=paste(age_cohort, sex, sep = "_")) %>% # variable to match change in mortality ra
    #   filter(cohort==age_sex_cohorts$cohort[i]) %>%
    #   left_join(mx_pylds_sc_total_disease_df_cohort) %>%
    #   mutate(rate=rate+replace_na(mortality_sum,0))%>%
    #   dplyr::select(-mortality_sum,-pylds_sum)



    suppressWarnings(
      general_life_table_list_sc[[i]] <- RunLifeTable(
        in_idata    = td3,
        in_sex      = age_sex_cohorts$sex[i],
        in_mid_age  = age_sex_cohorts$age[i],
        death_rates = NA
      ))
    names(general_life_table_list_sc)[i] <- age_sex_cohorts$cohort[i]
  }

  # convert the list of dataframes to single dataframes
  general_life_table_sc <- bind_rows(general_life_table_list_sc, .id = "age_group") %>%
    mutate(age_group = as.numeric(gsub("_.*","",age_group)))
```

Generate outputs data frame

```
  ## In the following list 'output_life_table', 34 data frames are nested per age and sex cohort
  ## Outputs are generated following the index order of disease life tables baseline and scenarios wher


#### Diseases life tables
  dia_index <- which(DISEASE_SHORT_NAMES$sname=='dmt2')
  dia_order <- c(dia_index,c(1:nrow(DISEASE_SHORT_NAMES))[-dia_index])

  ### Combine diseases and general life tables for scenarios
  ### Step needed to calculate numbers (rates*people cohort)
    scenario_d <- inner_join(disease_life_table_sc %>%
                             dplyr::select(age_group,sex,age,cause,incidence_disease,mx,px),
                           general_life_table_sc %>%
                             dplyr::select(age_group,sex,age,Lx,ex,Lwx,ewx),
                           by=c("age","sex","age_group")) %>%
    mutate(intervention="sc")

    baseline_d <- inner_join(disease_life_table_bl %>%
                             dplyr::select(age_group,sex,age,cause,incidence_disease,mx,px),
                           general_life_table_bl %>%
                             dplyr::select(age_group,sex,age,Lx,ex,Lwx,ewx),
                           by=c("age","sex","age_group")) %>%
    mutate(intervention="bl")

    disease_combined <- bind_rows(scenario_d,baseline_d) %>%
    pivot_wider(names_from  = intervention,
                values_from = c(incidence_disease,mx,px,Lx,ex,Lwx,ewx))  %>%
    mutate(inc_num_bl   = incidence_disease_bl*(1-px_bl)*Lx_bl,
           inc_num_sc   = incidence_disease_sc*(1-px_sc)*Lx_sc,
           inc_num_diff = inc_num_sc-inc_num_bl,
           mx_num_bl    = mx_bl*Lx_bl,
           mx_num_sc    = mx_sc*Lx_sc,
           mx_num_diff  = mx_num_sc-mx_num_bl) %>%
    dplyr::select(c(-mx_sc, -mx_bl, -px_sc, -px_bl, -Lx_sc, -Lx_bl, -ex_sc, -ex_bl, -Lwx_sc, -Lwx_bl, -e
      pivot_wider(names_from  = cause,
                values_from = inc_num_bl:mx_num_diff)


### Non_diseases life tables

    scenario_nd <- inner_join(non_disease_life_table_sc %>%
                             dplyr::select(age_group,sex,age,cause,mx,ylds_rate, diff_mort, diff_pylds)
                           general_life_table_sc %>%
                             dplyr::select(age_group,sex,age,Lx),
                           by=c("age","sex","age_group")) %>%
    mutate(intervention="sc")

    baseline_nd <- inner_join(non_disease_life_table_sc %>%
                             dplyr::select(age_group,sex,age,cause,mx,ylds_rate, diff_mort, diff_pylds)
                           general_life_table_bl %>%
                             dplyr::select(age_group,sex,age,Lx),
                           by=c("age","sex","age_group")) %>%
    mutate(intervention="bl")

    non_disease_combined <- bind_rows(scenario_nd,baseline_nd) %>%
    pivot_wider(names_from  = intervention,
                values_from = c(mx, ylds_rate, Lx,
                                values_fill=0)) %>%
    mutate(ylds_num_bl   = ylds_rate_bl*Lx_bl,
           ylds_num_sc   = ylds_rate_sc*Lx_sc,
           ylds_num_diff = ylds_num_sc-ylds_num_bl,
           mx_num_bl    = mx_bl*Lx_bl,
           mx_num_sc    = mx_sc*Lx_sc,
           mx_num_diff  = mx_num_sc-mx_num_bl) %>%
    dplyr::select(c(cause, sex, age, age_group, ylds_num_bl, ylds_num_sc, ylds_num_diff, mx_num_bl, mx_n
    pivot_wider(names_from  = cause,
                values_from = ylds_num_bl:mx_num_diff)

### General life tables

  general_lf <- bind_rows(
    general_life_table_sc %>%
      dplyr::select(age_group,sex,age,Lx,ex,Lwx,ewx) %>%
      mutate(intervention="sc"),
    general_life_table_bl %>%
      dplyr::select(age_group,sex,age,Lx,ex,Lwx,ewx) %>%
      mutate(intervention="bl")) %>%
    pivot_wider(names_from  = intervention,
                values_from = c(Lx,ex,Lwx,ewx),
                values_fill=0) %>%
    mutate(Lx_diff  = Lx_sc-Lx_bl,
           Lwx_diff = Lwx_sc-Lwx_bl,
           ex_diff  = ex_sc-ex_bl,
           ewx_diff = ewx_sc-ewx_bl)


  ######## Dataframe with all outputs by age and sex cohort over the simulation years (years of the coh
  output_df <- inner_join(disease_combined,
                          general_lf,
                          by=c("age","sex","age_group")) %>%
              inner_join(non_disease_combined,
                         by=c("age","sex","age_group"))

### Present changes in life expectancy, life years, health-adjusted life years and burden by cause (dis
```

Final function to use in mh-execute to run the PMSLT

```
source(paste0(relative_path_mslt, "R/RunMSLT.R"))

#### Graphs for changes in diseases over time
outputs <- RunMSLT(
  mslt_df=read_csv("C:\\Metahit\\mh-execute\\inputs\\mslt\\bristol_mslt.csv"),
  disease_names=readRDS("C:\\Metahit\\mh-mslt\\output\\parameters\\DISEASE_SHORT_NAMES.rds"),
  i_sex=c("male", "female"),
  i_age_cohort=seq(from=17, to=97, by =5),
  pif=read_csv("C:\\Metahit\\mh-mslt\\input\\pif.csv")
)
```

Examles of output

Tables

```
## Error in knitr::kable(outputs[["LifeYears"]], booktabs = T, caption = "Table 3: Life Years", : objec
```

```
## Error in knitr::kable(outputs[["LifeExpectancy"]], booktabs = T, caption = "Table 5: Life Expectancy
## Error in knitr::kable(outputs[["Cause"]], booktabs = T, caption = "Table 6: Diseases", : object ’outp
```

Graphs

```
## Error in eval(expr, envir, enclos): object ’outputs’ not found
```

```
## Error in eval(expr, envir, enclos): object ’outputs’ not found
```

```
## Error in dplyr::filter(output_df_agg_sex, sex == "female"): object ’output_df_agg_sex’ not found
```

```
## Error in dplyr::filter(output_df_agg_sex, sex == "male"): object ’output_df_agg_sex’ not found
```

```
## Error in eval(lhs, parent, parent): object ’output_df_agg_all’ not found
```

```
## Error in list2(...): object ’data_f’ not found
```

```
## Error:   You’re passing a function as global data.
##   Have you misspelled the ‘data‘ argument in ‘ggplot()‘
```

```
## Error in eval(expr, envir, enclos): object ’plot_1’ not found
```

```
## Error:   You’re passing a function as global data.
##   Have you misspelled the ‘data‘ argument in ‘ggplot()‘
```

```
## Error in eval(expr, envir, enclos): object ’plot_2’ not found
```

```
## Error in dplyr::filter(output_df_agg_sex, sex == "female"): object ’output_df_agg_sex’ not found
```

```
## Error in dplyr::filter(output_df_agg_sex, sex == "male"): object ’output_df_agg_sex’ not found
```

```
## Error in eval(lhs, parent, parent): object ’output_df_agg_all’ not found
```

```
## Error in eval(lhs, parent, parent): object ’data_t’ not found
```

```
## Error in eval(expr, envir, enclos): object ’plot_3’ not found
```