Contents

[1 General Principles 2](#_Toc158381245)

[2 Conceptual model 3](#_Toc158381246)

[3 Code Structure 4](#_Toc158381247)

[4 Data Inputs 5](#_Toc158381248)

[4.1 Populations/Population name.csv 5](#_Toc158381249)

[4.2 Parameter.rda 6](#_Toc158381250)

[4.3 LifeTables.csv 6](#_Toc158381251)

[5 User Defined Global Options 6](#_Toc158381252)

[5.1 Global Options.R 6](#_Toc158381253)

[6 Pre analysis functions 7](#_Toc158381254)

[6.1 R/build\_population.R 7](#_Toc158381255)

[6.2 R/generate\_random.R 7](#_Toc158381256)

[7 Script to run analyses 8](#_Toc158381257)

[7.1 Run all Analysis 8](#_Toc158381258)

[8 Analysis function 9](#_Toc158381259)

[8.1 R/run\_model.R 9](#_Toc158381260)

[9 Run a single model run 10](#_Toc158381261)

[9.1 R/run\_simulation.R 10](#_Toc158381262)

[10 Risk factor trajectories 12](#_Toc158381263)

[10.1 R/UKPDS 90 risk functions.R 12](#_Toc158381264)

[11 Intervention Effects 17](#_Toc158381265)

[11.1 R/intervention.R 17](#_Toc158381266)

[12 Create the matrix to store results 17](#_Toc158381267)

[12.1 R/Generate Results Template.R 17](#_Toc158381268)

[13 UKPDS 82 risk functions 18](#_Toc158381269)

[13.1 R/Update Events.R 18](#_Toc158381270)

[13.2 R/ UKPDS 82 risk functions.R 18](#_Toc158381271)

[13.3 R/LifeTableMortality.R 30](#_Toc158381272)

[14 UKPDS 90 risk functions 30](#_Toc158381273)

[14.1 R/Update Events.R 30](#_Toc158381274)

[14.2 R/UKPDS 90 risk functions.R 31](#_Toc158381275)

[15 Depression 34](#_Toc158381276)

[15.1 R/Update Events.R 34](#_Toc158381277)

[15.2 R/Depression.R 35](#_Toc158381278)

[16 Osteoarthritis 36](#_Toc158381279)

[16.1 R/Update Events.R 36](#_Toc158381280)

[16.2 R/ Oestoarthritis functions.R 36](#_Toc158381281)

[17 Cancer 37](#_Toc158381282)

[17.1 R/Update Events.R 37](#_Toc158381283)

[17.2 R/Cancer Risks.R 37](#_Toc158381284)

[18 Calculate QALYs 39](#_Toc158381285)

[18.1 R/ QALYs.R 39](#_Toc158381286)

[19 Calculate Costs 39](#_Toc158381287)

[19.1 R/Costs.R 39](#_Toc158381288)

[20 Record results 40](#_Toc158381289)

[20.1 R/Generate Results.R 40](#_Toc158381290)

[21 Update characteristics prior to moving to the next simulation year 41](#_Toc158381291)

[21.1 R/Update Pat Chars.R 41](#_Toc158381292)

[References 42](#_Toc158381293)

# General Principles

All inputs to a function have a trailing \_ at the end. Any derived values within the function lack the \_ at the end.

All risk functions from a single source, sit in their own .R file. If adding any new risk functions, you should make a new .R file. A new function should be written in apply\_events.R file to implement your new risk functions. The same principles apply to any equations for risk factor progression.

You should add a folder called Results to the model. Git ignores this file so it gives a logical place to save results of model runs without risking them being uploaded to GitHub. This folder can also be used to add results processing files in Excel if so wish.

# Conceptual model

The model is designed to extrapolate the typical outcomes of studies in people with type 2 diabetes (changes in: HbA1c; Body Mass Index (BMI); Systolic Blood Pressure (SBP); cholesterol) into changes in the expected incidence of complications related to diabetes. It also includes other events that are known to be related to weight (Depression, Breast Cancer, Colorectal Cancer and Osteoarthritis). The model is an individual simulation to allow tracking of complicated event histories.

Figure 1: The conceptual model underlying the model

A close-up of a diagram

Description automatically generated

# Code Structure

The structure of the model inputs and how they feed into the model functions are given below in Figure 1

Figure 2: The structure of the code in the model

A blue and white document with black text

Description automatically generated

# Data Inputs

## Populations/Population name.csv

This contains the people with diabetes characteristics for the simulation. Population name is user defined for your particular analysis. This is a raw data file that is processed using the build\_population function, which is located at (R/ build\_population.R file). The file located at data/PopulationVariables.csv lists all people with diabetes characteristics in the simulation and units for these characteristics.

Some default populations have been provided. These are:

|  |  |
| --- | --- |
| File name | Description |
| POPULATION1.csv | Population receiving 1st line pharmacotherapy in NICE NG28[1] |
| POPULATION2.csv | Population receiving 2nd line pharmacotherapy in NICE NG28[1] |
| POPULATION3.csv | Population receiving 3rd pharmacotherapy in NICE NG28[1] |
| POPULATION2\_earlyonset.csv | Population receiving 2nd line pharmacotherapy in NICE NG28, with resampling so only people with diabetes diagnosed before 40 are included[1] |
| POPULATION3.csv | Population receiving 3rd pharmacotherapy in NICE NG28, with resampling so only people with diabetes diagnosed before 40 are included[1] |
| POPULATION\_UKPDS.csv | Population matching most of the examples given in the UKPDS 82 and UKPDS 90 risk function papers, to allow unit tests[2, 3] |
| NICE, National Institute for Health and Care Excellence; UKPDS, UK Prospective Diabetes Study | |

**Note you should not use this model if you are using actual data from people with diabetes level (or bootstrapped people which is from a data set of people with diabetes) in your analysis. Any data put in this folder will be available under a GPL version 2 or later licence. It is therefore important to assign distributions to any people with diabetes level data you have and simulate characteristics from these distributions.**

## Parameter.rda

This is an RDA file that contains data on every parameter in the simulation. Each parameter is a separate column. The first row is the mean/median values for the parameter for a deterministic model run. The subsequent rows are random samples for a probabilistic model run.

This file is generated externally to the source code in this iteration of the model. Adaptations to this file should overwrite the existing Parameter.rda file

## LifeTables.csv

These are data extracted from the ONS lifetables. The format is a long format. As in the model AGE is in years and FEMALE is coded as 1 = female, 0 = male. Age 101 has been added with a probability of 1 to ensure that people die at age 101. These data are used within the model to ensure that the probability of death does not fall below the age-gender matched general population.

# User Defined Global Options

## Global Options.R

This is an R script that allows the user to set up a matrix that controls various options in the model, so that they don’t need to be hard coded or need multiple numbers defined.

The rownames are the user defined names of the columns

The 1st Column is Value, this contains the Value that you want this option to have. This can be numeric or character string. An example of a numeric option is n, which is 5000. This is to set the model to run 5000 people with diabetes. An example of a string is Results\_output, which, set as “Summary”, means that summary statistics from a model run will be reported and not the full people with diabetes characteristics matrix.

All options should be accompanied by a description and terms in this matrix should be used consistently throughout the model code.

This script returns a user defined matrix called GlobalVars, that can then be called into all model functions. Once made for the base case analysis, the GlobalVars matrix can be manipulated in the analysis file to allow the user to run different modelling scenarios. The only option in the matrix that does not need to be changed here is the treatment arm, as this is a direct user defined input into the functions that run the model analyses.

# Pre analysis functions

## R/build\_population.R

### build\_population

This function cleans the sampled population data and returns it in a format that is compatible with the model risk functions.

**Inputs**

diab\_diab\_population\_: this is the sampled population (normally be a populationname.csv file)

PopulationVariables\_: this is a list used to set variable names for the population matrix and key for all variables (normally be the Data/PopulationVariables.csv)

GlobalVars\_: this is a global matrix, set by the user, that is a list of options which will make the code run in different ways. Rows are named by the user, the 1st column is Value, this contains the number or string that is relevant to the user, the 2nd column is Description, which is written by user to allow people to understand what the option does.

In the current code for this function, the key option called “n”, which is the number of people with diabetes. This determines the number of rows of people with diabetes characteristics to produce.

**Output**

population: this is a processed matrix that contains people with diabetes characteristics in a format that allows the model to run.

## R/generate\_random.R

### generate\_random

This function returns an array of random numbers that cover 3 dimensions: the number of people with diabetes, the number of events that require random numbers, 100 cycles (proxy for maximum model run time). This array is called down through the model functions to ensure full variance reduction between model iterations when run probabilistically and between the model arms. This approach was taken to ensure common random numbers across probabilistic analysis when using parallel processing.so that random number differences do not cause a desync of random number attribution in different probabilistic analysis runs when parallel processing probabilistic analyses.

**Input**

n, this is the number of people with diabetes being run through these model iterations

**Output**

random\_numbers, this is an array of random numbers for use in the model that differs by people with diabetes, random event and year.

# Script to run analyses

## Run all Analysis

This script is the user defined script where you set the model to run all analyses within an analysis for your project. This calls in all other functions and data to some extent. Will save the results of each model iteration. This is not a function and needs to be rewritten from line 22 down for each analysis you do. If you want to change the population you are running lines 15 &16 will also need to be edited.

# Analysis function

## R/run\_model.R

### run\_model

This function runs the model given the inputs to the function. The model can run either deterministically or probabilistically. If run probabilistically, parallel processing across the probabilistic iterations of model parameters will be conducted to improve model run time.

**Inputs**

population\_, which is the processed population matrix

parameters\_, which is the whole parameters matrix

endtime\_, is a number specifying how many cycles the model should run. This allows you to set the time horizon to different lengths.

treatment\_, is a text term that is used to control what treatment scenario you wish to run through the model. This is user defined and is used by many functions that are called into run\_model.

GlobalVars\_, is a matrix of the global variables

random\_numbs\_, is the array of common random numbers

LifeTables\_ is the matrix of ONS data on Age, Gender and probability of death between that age and age+1 in a long format.

**Outputs**

Results, this is a matrix of results. This can be in many formats depending on a user specification. For example in a probabilistic model run this could be a limited set of results columns with a different row representing a different iteration of the probabilistic parameters. In a deterministic run, this could be more detailed set of summary results or the people with diabetes level matrix at the end of a model run. The output can be changed using the relevant option in the GlobalVars\_ matrix ()

# Run a single model run

## R/run\_simulation.R

### run\_simulation

This function runs the model once on a single set of parameters (it reduces the parameters matrix down to specified row). This code does everything, in the order specified below.

1. It reduces the parameters matrix down to the correct row. Row 1 is used if the model is run deterministically, or the probabilistic iteration +1 if the model is run probabilistically. The reason for this is that the first row of the parameters matrix is the means of the parameters and the subsequent rows are probabilistic iterations
2. It extrapolates the underlying risk factor trajectories according to the UKPDS90 equations over the entire model time horizon (Section ‎9)
3. It calculates the intervention effects relative to the underlying risk factor trajectories (note these functions should always be analysis specific) over the entire model time horizon (Section ‎10)
4. Initialises the matrix to store the results (Section ‎11)
5. Loop over the model years to determine the events
   1. Work out logical vectors of who is alive and dead in the population matrix
   2. Apply the model events in the UKPDS 82 risk equations (Section ‎12)
   3. Apply the events in the UKPDS 90 risk equations (Section ‎13)
   4. Apply the risk equation for whether or not someone develops depression (Section ‎14)
   5. Apply the risk equation for whether or not someone develops osteoarthritis (Section ‎15)
   6. Apply the risk for whether or not someone develops breast or colorectal cancer (Section ‎16)
   7. Unit test to make sure that people who where dead at the start of the year are not recorded as having an event & put all objects in the local environment into the global environment for bug checking
   8. Calculate QALYs (Section ‎17)
   9. Calculate Costs (Section ‎18)
   10. Produce detailed summary results (Section ‎19)
   11. Update people with diabetes characteristics and reset the correct variables so things such as costs do not carry over between years (Section ‎20)
   12. Unit test, if any event still exists (they should all now be recorded as a history of having that type of event), stop the model and put all objects in the local environment into the global environment
   13. Move to the next year
6. Return the results in your desired way, as specified in your functions and the GlobalVars\_ matrix

Inputs

population\_, is the population matrix

parameters\_, is the full parameters matrix

endtime\_, is the number of years to run the simulation for

treatment\_, is a text term that is used to control what treatment scenario you wish to run through the model. This is user defined and has many dependencies.

GlobalVars\_, is the matrix giving the global variables

random\_numbs\_, is an array of common random numbers giving a random number

LifeTables\_, is a dataframe containing life table information in format (long) that can easily be matched to the population matrix

SOUR\_, is the current second order uncertainty run (PSA iteration number if running a PSA, can be unspecified if running deterministically)

Outputs

results, is the results matrix

or

psaresults, is a summary set of results to produce in the PSA

# Risk factor trajectories

## R/UKPDS 90 risk functions.R

This file contains all risk functions that the model uses from UKPDS 90. All of these functions estimate the absolute values of each of their risk factors in absolute values. These functions use data from the UKPDS90 paper.[3]

If you wish to use different trajectories, these will need to be rewritten.

### UKPDS\_90\_contrisk\_A1c

This function estimates the HbA1c (in DCCT % scale) for all people with diabetes in the simulation over time based on their characteristics and on the risk equation given in UKPDS90, Supporting Information, file [dme14656-sup-0001-Appendix.docx](https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fdme.14656&file=dme14656-sup-0001-Appendix.docx), Table S3.[3]

Inputs:

population\_, is the population matrix

parameter\_, is a single row of the parameter matrix for this model run, specifically only the parameters relating to this equation are used

endtime\_, is the number of years to extrapolate HbA1c for

Outputs:

A1c, for each individual this gives the HBA1c (%) they would have in each future year of the simulation in the absence of treatment effects and death.

### UKPDS\_90\_contrisk\_BMI

This function estimates the Body Mass index (BMI) in Kg/m2 for all people with diabetes in the simulation over time based on their characteristics and on the risk equation given in UKPDS90, Supporting Information, file [dme14656-sup-0001-Appendix.docx](https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fdme.14656&file=dme14656-sup-0001-Appendix.docx), Table S3.[3]

Inputs:

population\_, is the population matrix

parameter\_, is a single row of the parameter matrix for this model run, specifically only the parameters relating to this equation are used

endtime\_, is the number of years to extrapolate BMI for

Outputs:

A1c, for each individual this gives the BMI (Kg/m2) they would have in each future year of the simulation in the absence of treatment effects and death.

### UKPDS\_90\_contrisk\_SBP

This function gives systolic blood pressure (SBP) in mm Hg, for all people with diabetes in the simulation over time based on their characteristics and on the risk equation given in UKPDS90, Supporting Information, file [dme14656-sup-0001-Appendix.docx](https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fdme.14656&file=dme14656-sup-0001-Appendix.docx), Table S3.[3]

Inputs:

population\_, is the population matrix

parameter\_, is a single row of the parameter matrix for this model run, specifically only the parameters relating to this equation are used

endtime\_, is the number of years to extrapolate SBP for

Outputs:

SBP, for each individual this gives the SBP (mm Hg) they would have in each future year of the simulation in the absence of treatment effects and death.

### UKPDS\_90\_contrisk\_HDL

This function gives High Density Lipoprotein (HDL) cholesterol in mmol/L, for all people with diabetes in the simulation over time based on their characteristics and on the risk equation given in UKPDS90, Supporting Information, file [dme14656-sup-0001-Appendix.docx](https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fdme.14656&file=dme14656-sup-0001-Appendix.docx), Table S3.[3]

Inputs:

population\_, is the population matrix

parameter\_, is a single row of the parameter matrix for this model run, specifically only the parameters relating to this equation are used

endtime\_, is the number of years to extrapolate HDL for

Outputs:

HDL, for each individual this gives the HDL (mmol/L) they would have in each future year of the simulation in the absence of treatment effects and death.

### UKPDS\_90\_contrisk\_LDL

This function gives Low Density Lipoprotein (LDL) cholesterol in mmol/L, for all people with diabetes in the simulation over time based on their characteristics and on the risk equation given in UKPDS90, Supporting Information, file [dme14656-sup-0001-Appendix.docx](https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fdme.14656&file=dme14656-sup-0001-Appendix.docx), Table S3.[3]

Inputs:

population\_, is the population matrix

parameter\_, is a single row of the parameter matrix for this model run, specifically only the parameters relating to this equation are used

endtime\_, is the number of years to extrapolate LDL for

Outputs:

LDL, for each individual this gives the LDL (mmol/L) they would have in each future year of the simulation in the absence of treatment effects and death.

### UKPDS\_90\_HEARTR

This functions gives the heart rate in beats per minute for all people with diabetes in the simulation over time based on their characteristics and on the risk equation given in UKPDS90, Supporting Information, file [dme14656-sup-0001-Appendix.docx](https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fdme.14656&file=dme14656-sup-0001-Appendix.docx), Table S3.[3] This is populated every 3 years, as the data in UKPDS 90 was collected every 3 years. Linear interpolation is used between the points every 3 years were heart rate values are estimated.

Inputs:

population\_, is the population matrix

parameter\_, is a single row of the parameter matrix for this model run, specifically only the parameters relating to this equation are used

endtime\_, is the number of years to extrapolate Heart Rate for

Outputs:

HEARTR, for each individual this gives the heart rate (beats per minute) they would have in each future year of the simulation in the absence of treatment effects and death.

### UKPDS\_90\_WBC

This functions gives the white blood cell count (WBC) in 1x106 ml for all people with diabetes in the simulation over time based on their characteristics and on the risk equation given in UKPDS90, Supporting Information, file [dme14656-sup-0001-Appendix.docx](https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fdme.14656&file=dme14656-sup-0001-Appendix.docx), Table S3.[3] This is populated every 3 years, as the data in UKPDS 90 was collected every 3 years. Linear interpolation is used between the points every 3 years were white blood cell count values are estimated.

Inputs:

population\_, is the population matrix

parameter\_, is a single row of the parameter matrix for this model run, specifically only the parameters relating to this equation are used

endtime\_, is the number of years to extrapolate Heart Rate for

Outputs:

WBC, for each individual this gives the WBC (1x106ml) they would have in each future year of the simulation in the absence of treatment effects and death.

### UKPDS\_90\_HAEM

This functions gives the haemoglobin in g/dL for all people with diabetes in the simulation over time based on their characteristics and on the risk equation given in UKPDS90, Supporting Information, file [dme14656-sup-0001-Appendix.docx](https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fdme.14656&file=dme14656-sup-0001-Appendix.docx), Table S3.[3] This is populated every 3 years, as the data in UKPDS 90 was collected every 3 years. Linear interpolation is used between the points every 3 years were white blood cell count values are estimated.

Inputs:

population\_, is the population matrix

parameter\_, is a single row of the parameter matrix for this model run, specifically only the parameters relating to this equation are used

endtime\_, is the number of years to extrapolate Heart Rate for

Outputs:

HAEM, for each individual this gives their haemoglobin (g/dL) they would have in each future year of the simulation in the absence of treatment effects and death.

# Intervention Effects

## R/intervention.R

These functions are project specific – so they should be rewritten for every project. The principle is that for each people with diabetes the change in each risk factor is explicitly recorded in a matrix. With each people with diabetes being a row, and the columns being years since baseline.

For further information, the paper for each model iteration should be read. A key dependency in these functions should always be the treatment\_ variable pulled through from the run\_simulation (Section ‎8.1.1) function, as this will allow this variable to control what HbA1c, BMI, SBP, HDL, LDL effects you are modelling.

# Create the matrix to store results

## R/Generate Results Template.R

This function initialises an empty matrix to populate when the model starts running over time. Each type of result stored from the population matrix is in the rows of the matrix and the number of years is across the columns.

**Inputs**

GlobalVar\_, Global variables matrix, written by the user

Endtime\_, is the model specified endtime. This determines the number of columns in the matrix

**Outputs**

results, is an empty matrix to populate from the loop when the actual model runs

# UKPDS 82 risk functions

## R/Update Events.R

### update\_events\_UKPDS82

This function calls in all of the UKPDS 82 risk functions that estimate the probability of having an event in the model: Chronic Heart Failure, Ischeamic Heart Disease, 1st Myocardial Infarction, 2nd Myocardial Infarction, 1st Stroke, 2nd Stroke, Blindness, Foot Ulcer, Amputation, 2nd Amputation, renal failure. It compares these probabilities to the array containing the common random numbers for this simulation to determine if each people with diabetes develops that complication or not. The random numbers depend on the individual with diabetes, the event and the year of the simulation.

**Inputs**

population\_, is the population matrix

parameters\_, is a single row of the parameters matrix

treatment\_, is the text option called all the way through the model indicating the treatment. This currently does nothing here. However, if you get an intervention with a Hazard Ratio, Odds Ratio, Relative Risk, this will allow you to apply these in the model functions as needed

Year\_, current simulation year

alive\_, logical vector. T = alive at the start of this year in the simulation. F = dead at the start of the year in the simulation

random\_numbs\_, array of common random numbers. Different numbers are generated for each person, event and year in the simulation

LifeTables\_, This is transformed input of the ONS lifetables

**Outputs**

population\_ , which is the revised population matrix with all events from UKPDS82 applied

## R/ UKPDS 82 risk functions.R

This folder contains all of the functions that implement the risk functions described in the electronic supplementary material (ESM) in UKPDS 82.[2]

### First\_CHF\_UKPDS\_82

This function estimates the probability that a people with diabetes develops a 1st Chronic Heart Failure event using the risk equation given in UKPDS 82, ESM Table 4.[2]

**Inputs**

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

rho\_, rho parameter for this risk equation

treatment\_, the user specified treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

pCHF, a vector of probabilities for every people with diabetes who is alive that they will have a 1st CHF

### First\_IHD\_UKPDS\_82

This function estimates the probability that a people with diabetes develops a 1st Ischemic Heart Disease event using the risk equation given in UKPDS 82, ESM Table 4.[2]

**Inputs**

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

rho\_, rho parameter for this risk equation

treatment\_, the user spefied treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

pIHD, a vector of probabilities for every people with diabetes who is alive that they will have a 1st IHD

### First\_MI\_UKPDS\_82

This function calls in the two functions below, depending on whether the person in the population is male or female. The function also ensures those with a history of MI are not simulated to have another first MI.

Inputs

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

rho\_f\_, rho parameter for this risk equation for women

treatment\_, the user spefied treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

Outputs

pMI, is the probability that each people with diabetes who was alive at the start of the year develops an MI this year.

#### First\_MI\_Male\_UKPDS\_82

This function estimates the probability that a male develops a 1st Myocardial Infarction event using the risk equation given in UKPDS 82, ESM Table 4.[2]

Inputs

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

treatment\_, the user spefied treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

Outputs

pMI, is the probability that each people with diabetes who was alive at the start of the year develops an MI this year according to the risk equation for males.

#### First\_MI\_Female\_UKPDS\_82

This function estimates the probability that a female develops a 1st Myocardial Infarction event using the risk equation given in UKPDS 82, ESM Table 4.[2]

Inputs

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

treatment\_, the user spefied treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

Outputs

pMI, is the probability that each people with diabetes who was alive at the start of the year develops an MI this year according to the risk equation for females.

### Second\_MI\_UKPDS\_82

This function estimates the probability that a people with diabetes develops a 2nd MI event (they must have had a 1st MI) using the risk equation given in UKPDS 82, ESM Table 4.[2]

Inputs

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

treatment\_, the user spefied treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

Outputs

pMI2, is the probability that each people with diabetes who was alive at the start of the year develops an 2nd MI this year

### First\_Stroke\_UKPDS\_82

This function estimates the probability that a people with diabetes develops a 1st Stroke event using the risk equation given in UKPDS 82, ESM Table 4.[2] The function ensures those with a history of stroke are not simulated to have another first stroke.

**Inputs**

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

rho\_, rho parameter for this risk equation

treatment\_, the user specified treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

pSTRO, a vector of probabilities for every people with diabetes who is alive that they will have a 1st stroke

### Second\_Stroke\_UKPDS\_82

This function estimates the probability that a people with diabetes develops a 2nd Stroke event using the risk equation given in UKPDS 82, ESM Table 4.[2] This is conditional on having a history of having had a stroke in previous years.

**Inputs**

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

rho\_, rho parameter for this risk equation

treatment\_, the user specified treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

pSTRO2, a vector of probabilities for every people with diabetes who is alive that they will have a 2nd stroke

### Blindness\_UKPDS\_82

This function estimates the probability that a people with diabetes develops blindness using the risk equation given in UKPDS 82, ESM Table 5.[2]

**Inputs**

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

rho\_, rho parameter for this risk equation

treatment\_, the user specified treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

pBLIND, a vector of probabilities for every people with diabetes who is alive that they will have become blind

### Ulcer\_UKPDS\_82

This function estimates the probability that a people with diabetes develops a foot ulcer using the risk equation given in UKPDS 82, ESM Table 5.[2]

**Inputs**

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

treatment\_, the user specified treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

pULCER, a vector of probabilities for every people with diabetes who is alive that they will have a foot ulcer

### First\_Amputation\_UKPDS\_82

This function calls in two subsequent functions to estimate the probability that someone has their 1st amputation depending on whether the people with diabetes has a history of foot ulcers or not. This is dependent on not already having had a 1st amputation.

**Inputs**

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

rho\_first\_amp\_noulcer\_, this is the rho parameter for the risk equation for 1st amputation for people with diabetes who do not have a history of foot ulcers

treatment\_, the user specified treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

pAMP, this is the probability that each people with diabetes has a 1st amputation

#### First\_Amputation\_Ulcer\_UKPDS\_82

This function estimates the probability that a people with diabetes develops an amputation if they have a history of having foot ulcers using the risk equation given in UKPDS 82, ESM Table 5.[2]

**Inputs**

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

treatment\_, the user specified treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Output**

pAMP, vector of probabilities of having an amputation for every people with diabetes that is included in the model

#### First\_Amputation\_noUlcer\_UKPDS\_82

This function estimates the probability that a people with diabetes develops an amputation if they do not have a history of having foot ulcers using the risk equation given in UKPDS 82, ESM Table 5.[2]

**Inputs**

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

rho\_first\_amp\_noulcer\_, is the rho parameter for this equation

treatment\_, the user specified treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Output**

pAMP, vector of probabilities of having an amputation for every people with diabetes that is included in the model

### Second\_Amputation\_UKPDS\_82

This function estimates the probability that a people with diabetes develops a 2nd amputation if they have a history of having previous amputations using the risk equation given in UKPDS 82, ESM Table 5.[2]

**Input**

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

treatment\_, the user specified treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Output**

pAMP2, is a vector of probabilities that each people with diabetes, who is alive in the given model year, has a 2nd amputation

### Renal\_UKPDS\_82

This function estimates the probability that a people with diabetes has renal failure using the risk equation given in UKPDS 82, ESM Table 5.[2]

**Input**

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

treatment\_, the user specified treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Output**

pRENAL, is a vector of probabilities for each person that is alive in the model that they develop renal failure this year.

### Death\_NoEvent\_Hist\_UKPDS\_82

This function estimates the probability of death predicted by UKPDS 82 data for people with diabetes who:

Have not had any of the following events this year: CHF, IHD, MI, 2nd MI, Stroke, 2nd Stroke, Amputation, 2nd Amputation, Renal Failure

And

Have a history of any of the following conditions: CHF, IHD, MI, 2nd MI, Stroke, 2nd Stroke, Amputation, 2nd Amputation, Renal Failure, Foot Ulcers, Blindness

The equation is given in UKPDS 82, ESM Table 6.[2]

**Inputs**

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

phi\_, is the phi parameter for this risk equation

treatment\_, the user specified treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

pDEATH, is a vector indicating the probability of death for every people with diabetes who is currently alive in the model

### Death\_NoEvent\_NoHist\_UKPDS\_82

This function estimates the probability of death predicted by UKPDS82 data for people with diabetes who:

Have not had any of the following events this year: CHF, IHD, MI, 2nd MI, Stroke, 2nd Stroke, Amputation, 2nd Amputation, Renal Failure

And

Do not have a history of any of the following conditions: CHF, IHD, MI, 2nd MI, Stroke, 2nd Stroke, Amputation, 2nd Amputation, Renal Failure, Foot Ulcers, Blindness

The equation is given in UKPDS 82, ESM Table 6.[2]

**Inputs**

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

phi\_, is the phi parameter for this risk equation

treatment\_, the user specified treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

pDEATH, is a vector indicating the probability of death for every people with diabetes who is currently alive in the model

### Death\_Event\_NoHist\_UKPDS\_82

This function estimates the probability of death predicted by the UKPDS82 data for people with diabetes who have at least one of the following events:

CHF, IHD, MI, 2nd MI, Stroke, 2nd Stroke, Amputation, 2nd Amputation, Renal Failure

And no history (events in previous years) of:

CHF, IHD, MI, 2nd MI, Stroke, 2nd Stroke, Amputation, 2nd Amputation, Renal Failure, Foot Ulcers, Blindness

Inputs

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

treatment\_, the user specified treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

Outputs

pDEATH, is a vector indicating the probability of death for every people with diabetes who is currently alive in the model

### Death\_Event\_Hist\_UKPDS\_82

This function estimates the probability of death predicted by the UKPDS82 data for people with diabetes who have at least one of the following events:

CHF, IHD, MI, 2nd MI, Stroke, 2nd Stroke, Amputation, 2nd Amputation, Renal Failure

And a history (events in previous years) of at least one of:

CHF, IHD, MI, 2nd MI, Stroke, 2nd Stroke, Amputation, 2nd Amputation, Renal Failure, Foot Ulcers, Blindness

**Inputs**

population\_, is the full population matrix

parameters\_, is a single row of the parameters matrix, corresponding to this probabilistic model run

treatment\_, the user specified treatment option (not used but does allow the application of Hazard Ratios, Relative Risks, Odds Ratios, if needed for your problem)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

pDEATH, is a vector indicating the probability of death for every people with diabetes who is currently alive in the model

## R/LifeTableMortality.R

### LifeTableMort

This function takes the age and gender of all people with diabetes in the model and matches it to data from the ONS life tables (transformed into a long format). This allows a comparison the probability of death from UKPDS 82 risk equations to ensure that people’s probability of death does not fall below that of someone in the general population

**Inputs**

population\_, is the full population matrix

LifeTables\_, This is the LifeTables.csv file pulled through the model (Section ‎3.3)

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

probs, a vector of probabilities associated with the age and gender of each people with diabetes in the population matrix.

# UKPDS 90 risk functions

## R/Update Events.R

### update\_events\_UKPDS90

This function updates all of the clinical events (Atrial Fibrillation, Peripheral Vascular Disease (PVD), Micro or Macro albuminuria), smoking histories and estimated glomerular filtration rate (eGFR) for people with diabetes in the model.

**Inputs**

population\_, the full population matrix

parameters\_, a single row of the parameters matrix

Year\_, current simulation year

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

random\_numbs\_, this is the common random number array generated in the analysis script (see Section ‎4.2.1)

**Outputs**

population\_, updated population matrix with all events from UKPDS 90 applied

## R/UKPDS 90 risk functions.R

### UKPDS\_90\_smo

This function updates the smoking status of people with diabetes in the model. As this data was collected every 3 years, this update is only done every three years in the model.

Details of the function are given in UKPDS90, Supporting Information, file [dme14656-sup-0001-Appendix.docx](https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fdme.14656&file=dme14656-sup-0001-Appendix.docx), Table S4.[3]

**Inputs**

population\_, is the full population matrix

parameter\_, is a single row of the parameter matrix

alive\_, is a logical vector. T = alive at the start of the year. F = dead at the start of the year

**Outputs**

prob\_smo, this is vector of probabilities that someone is a smoker for all people with diabetes who are alive

### UKPDS\_90\_MICALB

This function estimates the probability that each people with diabetes who is alive in the current year develops micro or macro albuminuria.

Details of the function are given in UKPDS90, Supporting Information, file [dme14656-sup-0001-Appendix.docx](https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fdme.14656&file=dme14656-sup-0001-Appendix.docx), Table S4.[3]

**Inputs**

population\_, is the full population matrix

parameter\_, is a single row of the parameters matrix

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Output**

prob\_MicAlb, a vector of probabilities for every people with diabetes that is alive that they develop Micro or Macroalbuminuria this year

### UKPDS\_90\_ATFIB

This function estimates the probability that each people with diabetes in the model develops Atrial Fibrillation in each year of the model.

Details of the function are given in UKPDS90, Supporting Information, file [dme14656-sup-0001-Appendix.docx](https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fdme.14656&file=dme14656-sup-0001-Appendix.docx), Table S4.[3]

**Inputs**

population\_, is the full population matrix

parameter\_, is a single row of the parameters matrix

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

prob\_ATFIB,

### UKPDS\_90\_PVD

This function estimates the probability that each people with diabetes in the model develops PVD in each year of the model.

Details of the function are given in UKPDS90, Supporting Information, file [dme14656-sup-0001-Appendix.docx](https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fdme.14656&file=dme14656-sup-0001-Appendix.docx), Table S4.[3]

**Inputs**

population\_, is the full population matrix

parameter\_, is a single row of the parameters matrix

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

prob\_PVD, this is the probability for all people with diabetes that are alive in a given year that they develop PVD this year

### UKPDS\_90\_binrary\_peGFRu60

This function estimates for each people with diabetes in the model whether their eGFR is below 60 ml/min/1.73m2 or not.

Details of the function are given in UKPDS90, Supporting Information, file [dme14656-sup-0001-Appendix.docx](https://onlinelibrary.wiley.com/action/downloadSupplement?doi=10.1111%2Fdme.14656&file=dme14656-sup-0001-Appendix.docx), Table S4.[3]

**Inputs**

population\_, is the full population matrix

parameter\_, is a single row of the parameters matrix

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

prob\_eGRFu60, a vector of probabilities that each person’s eGFR is below 60 (or not)

### UKPDS\_90\_eGFR

This function estimates for each people with diabetes in the model their eGFR level in ml/min/1.73m2 dependent on whether their eGFR was predicted to be over or under 60 ml/min/1.73m2.

**Inputs**

population\_, the full population matrix

parameter\_, a single row of the parameter matrix

eGRFu60\_, a vector which takes on a value of 0 or 1. 0 indicates that the corresponding person does not have an eGFR under 60. 1 indicates that their eGFR is under 60.

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

egfr, this is the egfr value corresponding to each people with diabetes in the simulation.

# Depression

## R/Update Events.R

### update\_events\_SPHR\_depression

This function is used to determine if each people with diabetes develops depression in a given year.

**Inputs**

population\_, this is the full population matrix

parameters\_, this is a single row of the parameters matrix

Year\_, this is the current simulation year

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

random\_numbs\_, this is the common random number array generated in the analysis script (see Section ‎4.2.1)

**Outputs**

population\_, updated population matrix

## R/Depression.R

### depression\_SPHR

This function implements the probability of depression as described in Breeze *et al*.[4] Key descriptions of the intention of this function are provided on pages 57-58.

**Inputs**

population\_, the full population matrix

parameters\_, a single row of the parameters matrix

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Output**

pDEP, probability for each people with diabetes in the model who was alive at the start of the year that they develop depression

# Osteoarthritis

## R/Update Events.R

### update\_events\_SPHR\_osteoarthritis

This function turns estimates the probability that someone develops osteoarthritis and estimates whether they develop a history of osteoarthritis or not.

**Inputs**

population\_, this is the full population matrix

parameters\_, this is a single row of the parameters matrix

Year\_, this is the current simulation year

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

random\_numbs\_, this is the common random number array generated in the analysis script (see Section ‎4.2.1)

**Outputs**

population\_, the updated population matrix including new cases of osteoarthritis

## R/ Oestoarthritis functions.R

This function implements the probability of developing osteoarthritis as described in Breeze *et al*.[4] Key descriptions of the intention of this function are provided on pages 56-57.

**Inputs**

population\_, the full population matrix

parameters\_, a single row of the parameters matrix

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Output**

pOST, probability for each people with diabetes in the model who was alive at the start of the year that they develop osteoarthritis in a given year.

# Cancer

## R/Update Events.R

### update\_events\_SPHR\_cancer

This function determines whether people with diabetes develop breast or colorectal cancer in a given simulation year.

**Inputs**

population\_, the full population matrix

parameter\_, a single row of the parameter matrix

Year\_, the current simulation year

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

random\_numbs\_, this is the common random number array generated in the analysis script (see Section ‎4.2.1)

**Outputs**

population\_,updated population matrix for the incidence of breast and colorectal cancer

## R/Cancer Risks.R

### Breast\_cancer

This function estimates the probability that each people with diabetes develops breast cancer in a given model year, as described in Breeze *et al*.[4] Key descriptions of the intention of this function are provided on pages 55-56.

**Inputs**

population\_, the full population matrix

parameters\_, a single row of the parameters matrix

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

pBC, probability that each people with diabetes who is alive at the start of the simulation year develops breast cancer

### Colorectal\_cancer

This function estimates the probability that each people with diabetes develops breast cancer in a given model year, as described in Breeze *et al*.[4] Key descriptions of the intention of this function are provided on pages 56.

**Inputs**

population\_, the full population matrix

parameters\_, a single row of the parameters matrix

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

**Outputs**

pCC, probability for each people with diabetes who is alive, that they develop colorectal cancer in a given year

# Calculate QALYs

## R/ QALYs.R

### calculate\_QALYs

Calculates QALYs for people with diabetes given the events that happen this year. For people who survive the year, this is assumed to be a full year for people who died this year their death is assumed to occur halfway through the year.

Details will be available in forthcoming publications

**Inputs**

population\_, the full population matrix

parameters\_, a single row of the parameters matrix

year\_, the current simulationyear

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

GlobalVars\_, this is the matrix of Global Options set by the user

**Outputs**

population\_, this is the population matrix that has recorded the estimated QALYs and discounted QALYs for this year

# Calculate Costs

## R/Costs.R

### calculate\_costs

Calculates costs for people with diabetes given the events that happen this year. For people who survive the year, this is assumed to be a full year for people who died this year their death is assumed to occur halfway through the year.

Details will be available in forthcoming publications

**Inputs**

population\_, the full population matrix

parameters\_, a single row of the parameters matrix

year\_, the current simulation year

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

GlobalVars\_, this is the matrix of Global Options set by the user

treatment\_, this is the text term put into the run model function to indicate which treatment arm is been run

attend\_se\_, this is a matrix that is intervention specific for the Embedding analyses. This only applies in the scenario where the Embedding Step Wedge study was only used to estimate the attendance at SE.

**Outputs**

population\_, this is the updated population matrix including discounted and undiscounted costs accrued in a single simulation year

# Record results

## R/Generate Results.R

### GenerateDetailedresults

This function summarises the population matrix for key variables that accrue in each year. This is mostly sums of events, histories, costs and QALYs. But also includes mean risk factor values. Additional variables can be added as needed to the simulation for the validation needs of the analyst.

**Inputs**

results\_, a semi-complete results matrix which includes the results for all previous simulation year

population\_, this is the population matrix

year\_, this is the simulation year

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

GlobalVars\_, this is the user defined global options for this model run

**Outputs**

results\_, results matrix, with the summaries of this years results completed

# Update characteristics prior to moving to the next simulation year

## R/Update Pat Chars.R

### update\_history

This function updates the events to being histories of events, resets EQ5D and QALY columns in the population matrix, sets HbA1c, BMI, SBP, HDL, LDL, Heart rate, WBC and Haemoglobin to the value they should take next year for the people with diabetes who survived this year of the simulation. This is adding the trajectory to any user specified treatment effects for each people with diabetes in the model.

**Inputs**

population\_, the full population matrix

HBA1c\_underlying\_, is the HbA1c trajectory for each people with diabetes for each simulation year estimated from UKPDS90 (see Section ‎9.1.1)

BMI\_underlying\_, is the BMI trajectory for each people with diabetes for each simulation year from UKPDS 90 (see Section‎ 9.1.2)

SBP\_underlying\_, is the systolic blood pressure trajectory for each people with diabetes for each simulation year from UKPDS90 (see Section ‎9.1.3)

HDL\_underlying\_, is the HDL cholesterol trajectory for each people with diabetes for each simulation year form UKPDS90 (see Section ‎9.1.4)

LDL\_underlying\_, is the LDL cholesterol trajectory for each people with diabetes for each simulation year from UKPDS90 (see Section ‎9.1.5)

HEARTR\_underlying\_, is the heart rate trajectory for each people with diabetes for each simulation year from UKPDS90 (see Section ‎9.1.6)

WBC\_underlying\_, is the white blood cell count trajectory for each people with diabetes for each simulation year from UKPDS90 (see Section ‎9.1.7)

HAEM\_underlying\_, is the heamoglobin trajectory for each people with diabetes for each simulation year from UKPDS90 (see Section ‎9.1.8)

HBA1c\_INTV\_, is the matrix of user specified changes in HbA1c from the trajectory for this particular model run. This is usually where intervention effects are applied

BMI\_INTV\_, is the matrix of user specified changes in BMI from the trajectory for this particular model run. This is usually where intervention effects are applied

SBP\_INTV\_, is the matrix of user specified changes in SBP from the trajectory for this particular model run. This is usually where intervention effects are applied

HDL\_INTV\_, is the matrix of user specified changes in HDL from the trajectory for this particular model run. This is usually where intervention effects are applied

LDL\_INTV\_, is the matrix of user specified changes in LDL from the trajectory for this particular model run. This is usually where intervention effects are applied

year\_, current simulation year

**Outputs**

population\_, updated population matrix including updated event histories, and risk factor levels for each people with diabetes who survived this simulation year.

### update\_patchars

This function updates people with diabetes age, diabetes duration, years lived, probability of being a smoker, whether they have menopause, and what diabetes therapy they receive.

Inputs

population\_, this is the full population matrix

parameters\_, this is a single row of the parameters matrix

alive\_, this is logical vector generated in the run\_simulation function, T if a people with diabetes is alive at the start of the year, F otherwise

# References

1. National Institute for Health and Care Excellence. Type 2 diabetes in adults: management. 2015.

2. Hayes AJ, Leal J, Gray AM, Holman RR, Clarke PM. UKPDS outcomes model 2: a new version of a model to simulate lifetime health outcomes of patients with type 2 diabetes mellitus using data from the 30 year United Kingdom Prospective Diabetes Study: UKPDS 82. *Diabetologia* 2013; **56**:1925-1933.

3. Leal J, Alva M, Gregory V, Hayes A, Mihaylova B, Gray AM*, et al.* Estimating risk factor progression equations for the UKPDS Outcomes Model 2 (UKPDS 90). *Diabet Med* 2021; **38**:e14656.

4. Breeze PR, Thomas C, Squires H, Brennan A, Greaves C, Diggle PJ*, et al.* School for Public Health Research (SPHR) Diabetes Prevention Model: Detailed Description of Model Background, Methods, Assumptions and Parameters. *HEDS discussion paper No1501* 2015.