

ITHIM-R

August 6, 2019

`add_distance_columns` *Add distance columns to injury tables*

Description

Add distance columns to injury tables, matching on information in the injury contingency table

Usage

```
add_distance_columns(injury_table, mode_names, true_distances_0, dist,  
                     scenarios = SCEN)
```

Arguments

<code>injury_table</code>	(list of) data frame(s) to be edited
<code>mode_names</code>	which modes to take distances for
<code>true_distances_0</code>	distances to add to injury table
<code>dist</code>	table used to access bus distance
<code>scenarios</code>	which scenarios to process

Value

edited (list of) data frame(s)

add_ghost_trips	<i>Add trips taken by no one</i>
-----------------	----------------------------------

Description

Creates trips based on distance relative to another (reference) mode, without altering the synthetic population

Usage

```
add_ghost_trips(raw_trip_set, trip_mode = "bus_driver",
  distance_ratio = BUS_TO_PASSENGER_RATIO * DISTANCE_SCALAR_PT,
  reference_mode = "bus")
```

Arguments

- raw_trip_set data frame of trips
- trip_mode which mode to add
- distance_ratio fraction of reference distance to create
- reference_mode name of reference mode

Value

data frame of trips

add_trips	<i>Add trips to trip set</i>
-----------	------------------------------

Description

Creates a data frame of the same description as a trip set to append

Usage

```
add_trips(trip_ids = 0, new_mode = "walking", distance = 1,
  participant_id = 0, age = 20, sex = "Male", nTrips = 3,
  speed = 4.8)
```

Arguments

trip_ids	ids for new trips
new_mode	mode for new trips
distance	distances to sample from
participant_id	participant id for new trips
age	age for participant
sex	sex for participant
nTrips	number of trips for participant
speed	speed for new trips

Value

data frame of trips

add_walk_trips	<i>Add walk trips to trip set</i>
----------------	-----------------------------------

Description

Create data frame of walk-to-PT trips from PT trips and walk-to-bus time

Usage

```
add_walk_trips(pt_trips)
```

Arguments

pt_trips	data frame of PT trips
----------	------------------------

Value

list of data frames of PT trips and walk-to-PT trips

ap_dose_response_curve

Computes RR as a DR relationship

Description

Computes RR as a DR relationship given four parameters and the PM2.5 exposure

Usage

```
ap_dose_response_curve(pm, alpha, beta, gamma, tmrel)
```

Arguments

pm	PM2.5 exposure
alpha	DR parameter
beta	DR parameter
gamma	DR parameter
tmrel	DR parameter

Value

RR

assign_age_groups

Assign age groups to individuals

Description

Prunes dataset given max and min ages; assigns age group labels given age

Usage

```
assign_age_groups(dataset, age_category = AGE_CATEGORY,
  age_lower_bounds = AGE_LOWER_BOUNDS, max_age = MAX_AGE,
  min_age = AGE_LOWER_BOUNDS[1], age_label = "age")
```

Arguments

dataset	data frame to be edited
age_category	vector of strings giving age categories
age_lower_bounds	lower boundaries of age categories
max_age	maximum age for model
min_age	minimum age for model
age_label	string label for age column

Value

edited data frame

beta_pointiness	<i>Parametrise confidence in PA data</i>
-----------------	--

Description

Takes a confidence value between 0 and 1 and returns a parameter for a beta distribution

Usage

```
beta_pointiness(confidence)
```

Arguments

confidence value between 0 and 1 representing how confident we are about the PA dataset

Value

a value to parametrise a beta distribution

combined_rr_ap_pa	<i>Combine relative risks from AP and PA</i>
-------------------	--

Description

Combine relative risks from AP and PA through multiplication for crossover diseases

Usage

```
combined_rr_ap_pa(ind_pa, ind_ap)
```

Arguments

ind_pa data frame of individual RRs for diseases affected by PA
ind_ap data frame of individual RRs for diseases affected by AP

Value

combined RR for diseases after accounted for AP and PA exposures

`combine_health_and_pif`*Combine health and PIF*

Description

Applies PIF calculated from RRs to the current observed health burden from GBD to generate scenario health burden

Usage

```
combine_health_and_pif(pif_values, hc = DISEASE_BURDEN)
```

Arguments

<code>pif_values</code>	vector of values of PIFs
<code>hc</code>	data frame of current burden of disease

Value

estimated scenario burden of disease

`complete_trip_distance_duration`*Data harmonisation: add columns to trip set if missing*

Description

Creates any columns needed and missing from the trip set, then writes the trip set to the global environment

Usage

```
complete_trip_distance_duration()
```

create_all_scenarios	<i>Creates specific scenarios for Accra and Sao Paulo</i>
----------------------	---

Description

Creates five prespecified scenarios from the baseline for Accra and Sao Paulo

Usage

```
create_all_scenarios(trip_set)
```

Arguments

trip_set	data frame of baseline trips
----------	------------------------------

Value

list of scenarios

create_cycle_scenarios	<i>Creates cycling scenarios</i>
------------------------	----------------------------------

Description

Creates five scenarios with 10-50

Usage

```
create_cycle_scenarios(trip_set)
```

Arguments

trip_set	data frame of baseline trips
----------	------------------------------

Value

list of scenarios

```
create_max_mode_share_scenarios
```

Create scenarios defined by maximum mode share

Description

Creates five scenarios where, in each one, the mode share is elevated to the maximum observed across the cities. The scenario-modes are walking, cycling, car, motorcycle and bus

Usage

```
create_max_mode_share_scenarios(trip_set)
```

Arguments

trip_set data frame, baseline scenario

Value

list of baseline scenario and five mode scenarios

```
create_scenario
```

Create individual scenario

Description

Function to create individual scenario for the five prespecified scenarios from the baseline for Accra and Sao Paulo (create_all_scenarios)

Usage

```
create_scenario(rdr, scen_name, source_modes, combined_modes = F,
               target_modes, source_distance_cats, source_trips, target_trips)
```

Arguments

rdr data frame of trips
 scen_name name of scenario
 source_modes which mode(s) to take trips from
 combined_modes whether or not to combine source modes
 target_modes mode to change to
 source_distance_cats which categories to select trips from
 source_trips how many trips to leave, or to take
 target_trips

Value

list of scenarios

create_synth_pop	<i>Creates synthetic population</i>
------------------	-------------------------------------

Description

Creates a synthetic population by matching individuals in the trip set to individuals in the PA set

Usage

```
create_synth_pop(raw_trip_set)
```

Arguments

raw_trip_set data frame of raw trips taken

Value

the synthetic population and the trip set which has been pruned

create_walk_scenario	<i>Create basic walk scenario</i>
----------------------	-----------------------------------

Description

Duplicate baseline scenario; give each person one 1km walk in the scenario

Usage

```
create_walk_scenario(trip_set)
```

Arguments

trip_set data frame, baseline scenario

Value

list of baseline scenario and walking scenario

`dirichlet_pointiness` *Function for Dirichlet parameters*

Description

Function to map a confidence value to a parametrisation of a Dirichlet distribution

Usage

```
dirichlet_pointiness(confidence)
```

Arguments

`confidence` value between 0 and 1

Value

parametrisation

`distances_for_injury_function`
Get distances and model for injuries module

Description

Computes exposures (distances) to parametrise the injury regression model, which is computed as a Poisson with various offsets and used later in prediction

Usage

```
distances_for_injury_function(trip_scen_sets, dist)
```

Arguments

`trip_scen_sets` list of synthetic trip sets for scenarios
`dist` table of (total) distances per mode per scenario

Value

list of distances, injury table, and injury regression model

dist_dur_tbls	<i>Get distances and durations</i>
---------------	------------------------------------

Description

Summaries of total distances and durations spent travelling per mode per scenario

Usage

```
dist_dur_tbls(trip_scen_sets)
```

Arguments

trip_scen_sets list of synthetic trip sets for scenarios

Value

list of table of (total) distances and durations per mode per scenario

gen_ap_rr	<i>Get RR for diseases given AP exposure</i>
-----------	--

Description

Computes the RR for individuals for each disease given AP exposure

Usage

```
gen_ap_rr(pm_conc_pp)
```

Arguments

pm_conc_pp individual AP exposures

Value

data frame of relative risks per person per disease

gen_pa_rr	<i>Get RR for diseases given mMETs</i>
-----------	--

Description

Computes the RR for individuals for each disease given mMETs (PA exposure)

Usage

```
gen_pa_rr(mmets_pp)
```

Arguments

mmets_pp individual mMETs

Value

data frame of relative risks per person per disease

get_all_distances	<i>Sequence to get distance data</i>
-------------------	--------------------------------------

Description

Sequence of function calls to get distance data for modules from synthetic population

Usage

```
get_all_distances(ithim_object)
```

Arguments

ithim_object list containing synthetic trip set

Value

ithim_object again, with additional distance objects

get_scenario_settings *Get values for max mode share scenario*

Description

Computes the maximum mode share for specified mode types and specified distance categories across specified (stored) cities. Used for max mode share scenario generation.

Usage

```
get_scenario_settings(cities = c("accra", "sao_paulo", "delhi",
  "bangalore"), modes = c("walking", "bicycle", "car", "motorcycle",
  "bus"), distances = c("0-1 km", "2-5 km", "6+ km"), speeds = list(bus
  = 15, bus_driver = 15, minibus = 15, minibus_driver = 15, car = 21, taxi
  = 21, walking = 4.8, walk_to_pt = 4.8, bicycle = 14.5, motorcycle = 25,
  truck = 21, van = 15, subway = 28, rail = 35, auto_rickshaw = 22,
  shared_auto = 22, cycle_rickshaw = 10))
```

Arguments

cities	which cities to use
modes	which modes to use
distances	which distance categories to use
speeds	named list of mode speeds (to be applied to all cities)

Value

data frame of proportions by mode and distance category

get_synthetic_from_trips

Generate synthetic data from trip set

Description

Sequence of functions to set up the synthetic population, the synthetic trips, and the scenarios. Also sets global variables for later use.

Usage

```
get_synthetic_from_trips()
```

Value

data frame of all trips from all scenarios

health_burden	<i>Compute health burden</i>
---------------	------------------------------

Description

Compute health burden for populations in scenarios given relative risks

Usage

```
health_burden(ind_ap_pa, combined_AP_PA = T)
```

Arguments

ind_ap_pa data.frame of all individuals' relative risks for diseases
 combined_AP_PA=T
 logic: whether to combine the two exposure pathways (AP and PA) or to compute independently

Value

list of data.frames: one for deaths per disease per demographic group, and likewise for YLLs

injuries_function_2	<i>Predict injuries</i>
---------------------	-------------------------

Description

Predict injuries based on regression model from baseline and scenario travel

Usage

```
injuries_function_2(true_distances, injuries_list, reg_model,  
                    constant_mode = F)
```

Arguments

true_distances data frame to set up results
 injuries_list list of data frames to supply to regression model for prediction
 reg_model regression glm object
 constant_mode whether or not we are in constant (vs sampling) mode

Value

list of injury prediction data frames and whw matrices

injury_death_to_yll	<i>Map injury death burden to YLL burden</i>
---------------------	--

Description

Calculated the YLL burden from the death burden of injury based on the ratio in the GBD data.

Usage

```
injury_death_to_yll(injuries)
```

Arguments

injuries	data frame of injury deaths
----------	-----------------------------

Value

list of injury deaths and YLLs (which are differences from reference scenario) plus the values in the reference scenario.

ithim_calculation_sequence	<i>Cascade of computations that form the ITHIM</i>
----------------------------	--

Description

Ordered set of computations that form the ITHIM, from travel information to health burden.

Usage

```
ithim_calculation_sequence(ithim_object, seed = 1)
```

Arguments

ithim_object	name of disease
seed	

Value

list of items making up the ithim result

ithim_load_data	<i>Load data for model and setting</i>
-----------------	--

Description

Loads and processes data from file. Local data for the setting and global data for the model. Writes objects to the global environment.

Usage

```
ithim_load_data(speeds = list(bus = 15, bus_driver = 15, minibus = 15,  
  minibus_driver = 15, car = 21, taxi = 21, walking = 4.8, walk_to_pt =  
  4.8, bicycle = 14.5, motorcycle = 25, truck = 21, van = 15, subway = 28,  
  rail = 35, auto_rickshaw = 22, shared_auto = 22, cycle_rickshaw = 10))
```

Arguments

speeds	named list of mode speeds
--------	---------------------------

ithim_setup_baseline_scenario	<i>Set up baseline scenario data frame</i>
-------------------------------	--

Description

Create scenario by adding distance categories and scenario=baseline column to trip set data frame

Usage

```
ithim_setup_baseline_scenario(trip_set)
```

Arguments

trip_set	data frame of trips
----------	---------------------

Value

trip_set as baseline scenario

ithim_setup_parameters

Routine to sample or set parameters for ITHIM

Description

Parameters have two options: to be set to a constant, and to be sampled from a prespecified distribution. Each parameter is given as an argument of length 1 or 2. If length 1, it's constant, and set to the global environment. If length 2, a distribution is defined and sampled from NSAMPLE times. There are some exceptions, listed below.

Usage

```
ithim_setup_parameters(NSAMPLES = 1, BUS_WALK_TIME = 5,
  MMET_CYCLING = 4.63, MMET_WALKING = 2.53, PM_CONC_BASE = 50,
  PM_TRANS_SHARE = 0.225, PA_DOSE_RESPONSE_QUANTILE = F,
  AP_DOSE_RESPONSE_QUANTILE = F, BACKGROUND_PA_SCALAR = 1,
  BACKGROUND_PA_CONFIDENCE = 1, INJURY_REPORTING_RATE = 1,
  CHRONIC_DISEASE_SCALAR = 1, DAY_TO_WEEK_TRAVEL_SCALAR = 7,
  INJURY_LINEARITY = 1, CASUALTY_EXPONENT_FRACTION = 0.5,
  BUS_TO_PASSENGER_RATIO = 0.022, TRUCK_TO_CAR_RATIO = 0.21,
  EMISSION_INVENTORY_CONFIDENCE = 1, DISTANCE_SCALAR_CAR_TAXI = 1,
  DISTANCE_SCALAR_WALKING = 1, DISTANCE_SCALAR_PT = 1,
  DISTANCE_SCALAR_CYCLING = 1, DISTANCE_SCALAR_MOTORCYCLE = 1)
```

Arguments

NSAMPLES	constant integer: number of samples to take
BUS_WALK_TIME	lognormal parameter: duration of walk to PT
MMET_CYCLING	lognormal parameter: mMETs when cycling
MMET_WALKING	lognormal parameter: mMETs when walking
PM_CONC_BASE	lognormal parameter: background PM2.5 concentration
PM_TRANS_SHARE	beta parameter: fraction of background PM2.5 attributable to transport
PA_DOSE_RESPONSE_QUANTILE	logic: whether or not to sample from PA RR DR functions
AP_DOSE_RESPONSE_QUANTILE	logic: whether or not to sample from AP RR DR functions
BACKGROUND_PA_SCALAR	lognormal parameter: reporting scalar for PA
BACKGROUND_PA_CONFIDENCE	beta parameter: confidence in accuracy of PA survey
INJURY_REPORTING_RATE	lognormal parameter: rate of injury reporting
CHRONIC_DISEASE_SCALAR	lognormal parameter: scalar for background disease rates

DAY_TO_WEEK_TRAVEL_SCALAR
 beta parameter: rate of scaling travel from one day to one week

INJURY_LINEARITY
 lognormal parameter: linearity of injuries in space

CASUALTY_EXPONENT_FRACTION
 beta parameter: casualty contribution to linearity of scaling of injuries in space

BUS_TO_PASSENGER_RATIO
 beta parameter: number of buses per passenger

TRUCK_TO_CAR_RATIO
 beta parameter: number of trucks per car

EMISSION_INVENTORY_CONFIDENCE
 beta parameter: confidence in accuracy of emission inventory

DISTANCE_SCALAR_CAR_TAXI
 lognormal parameter: scalar for car distance travelled

DISTANCE_SCALAR_WALKING
 lognormal parameter: scalar for walking distance travelled

DISTANCE_SCALAR_PT
 lognormal parameter: scalar for PT distance travelled

DISTANCE_SCALAR_CYCLING
 lognormal parameter: scalar for cycling distance travelled

DISTANCE_SCALAR_MOTORCYCLE
 lognormal parameter: scalar for motorcycle distance travelled

Value

list of samples of uncertain parameters

ithim_uncertainty	<i>Sampling routine for calculating ITHIM with uncertainty</i>
-------------------	--

Description

Sets sampled parameters to the global environment and runs the ITHIM routine (run_ithim)

Usage

```
ithim_uncertainty(ithim_object, seed = 1)
```

Arguments

ithim_object	list of necessary inputs, including parameters
seed	which sample to take

Value

list of ITHIM outcomes

join_hb_and_injury	<i>Join disease health burden and injury</i>
--------------------	--

Description

Join the two data frames for health burden: that from disease, and that from road-traffic injury

Usage

```
join_hb_and_injury(ind_ap_pa, inj)
```

Arguments

ind_ap_pa	list (deaths, YLLs) of data frames of all demographic groups' burdens for diseases
inj	list (deaths, YLLs) of data frames of all demographic groups' burdens for road-traffic injury

Value

list of data.frames: one for deaths per cause per demographic group, and likewise for YLLs

PA_dose_response	<i>Calculate RR given PA</i>
------------------	------------------------------

Description

Calculate RR for a disease given PA

Usage

```
PA_dose_response(cause, dose, confidence_intervals = F)
```

Arguments

cause	name of disease
dose	vector of doses of PA from individuals
confidence_intervals	logic: whether or not to return confidence intervals

Value

data frame of relative risks

`parallel_evppi_for_AP` *Compute AP EVPPI in parallel*

Description

Compute AP EVPPI in parallel

Usage

```
parallel_evppi_for_AP(disease, parameter_samples, outcome, NSCEN)
```

Arguments

disease	disease name
parameter_samples	data frame of parameter samples
outcome	data frame of outcomes
NSCEN	number of scenarios

Value

vector of EVPPI values (one per scenario)

`population_attributable_fraction`
Calculate population attributable fraction

Description

Calculate population attributable fraction

Usage

```
population_attributable_fraction(pop, cn, mat)
```

Arguments

pop
cn
mat

Value

population attributable fractions by demographic group

run_ithim	<i>Wrapper for running ITHIM</i>
-----------	----------------------------------

Description

Switch to run the computation directly, or divert to the sampling case

Usage

```
run_ithim(ithim_object, seed = 1)
```

Arguments

ithim_object list of items making up the ithim set up
seed

Value

ithim_object list of items making up the ithim result

run_ithim_setup	<i>Run the set up script for ITHIM</i>
-----------------	--

Description

Sets up the basic ITHIM object for onward calculation. Data loading, processing and harmonisation. Setting global values.

Usage

```
run_ithim_setup(seed = 1, CITY = "accra", speeds = NULL,
  emission_inventory = NULL,
  setup_call_summary_filename = "setup_call_summary.txt",
  DIST_CAT = c("0-6 km", "7-9 km", "10+ km"), AGE_RANGE = c(0, 150),
  ADD_WALK_TO_BUS_TRIPS = T, ADD_BUS_DRIVERS = T,
  ADD_TRUCK_DRIVERS = T, TEST_WALK_SCENARIO = F,
  TEST_CYCLE_SCENARIO = F, MAX_MODE_SHARE_SCENARIO = F,
  REFERENCE_SCENARIO = "Baseline", PATH_TO_LOCAL_DATA = NULL,
  NSAMPLES = 1, BUS_WALK_TIME = 5, MMET_CYCLING = 4.63,
  MMET_WALKING = 2.53, PM_CONC_BASE = 50, PM_TRANS_SHARE = 0.225,
  PA_DOSE_RESPONSE_QUANTILE = F, AP_DOSE_RESPONSE_QUANTILE = F,
  BACKGROUND_PA_SCALAR = 1, BACKGROUND_PA_CONFIDENCE = 1,
  INJURY_REPORTING_RATE = 1, CHRONIC_DISEASE_SCALAR = 1,
  DAY_TO_WEEK_TRAVEL_SCALAR = 7, INJURY_LINEARITY = 1,
  CASUALTY_EXPONENT_FRACTION = 0.5, MOTORCYCLE_TO_CAR_RATIO = 0.2,
```

```
BUS_TO_PASSENGER_RATIO = 0.022, TRUCK_TO_CAR_RATIO = 0.21,  
EMISSION_INVENTORY_CONFIDENCE = 1, DISTANCE_SCALAR_CAR_TAXI = 1,  
DISTANCE_SCALAR_WALKING = 1, DISTANCE_SCALAR_PT = 1,  
DISTANCE_SCALAR_CYCLING = 1, DISTANCE_SCALAR_MOTORCYCLE = 1)
```

Value

ithim_object list of objects for onward use.

scale_trip_distances	<i>Scale trip distances</i>
----------------------	-----------------------------

Description

Applies mode-specific distance scalars to all trips

Usage

```
scale_trip_distances(trips)
```

Arguments

trips data frame, all trips from all scenarios

Value

data frame, all trips from all scenarios

scenario_pm_calculations	<i>Calculate total AP exposure per person</i>
--------------------------	---

Description

Calculate total AP exposure per person based on population and personal travel

Usage

```
scenario_pm_calculations(dist, trip_scen_sets)
```

Arguments

dist data frame of population travel from all scenarios
trip_scen_sets data frame of all trips from all scenarios

Value

background AP
total AP exposure per person

set_injury_contingency

Create contingency table from itemised list of injuries

Description

One of the inputs is a list of injury events. This function aggregates injuries by type into a long contingency table with prespecified column names. Write tables to global environment.

Usage

```
set_injury_contingency(injuries)
```

Arguments

injuries data frame of injury events

set_vehicle_inventory *Collate all vehicle information*

Description

Puts all vehicle information in one place. Writes to global environment.

Usage

```
set_vehicle_inventory()
```

summarise_ithim_inputs

Graphical processing of input data

Description

Produce graphs summarising some input data, e.g. travel, injury, AP

Usage

```
summarise_ithim_inputs(ithim_object)
```

Arguments

ithim_object processed ithim_object from run_ithim_setup

total_mmet	<i>Calculate total mMETs per person</i>
------------	---

Description

Calculate total mMETs per person based on PA and active travel

Usage

```
total_mmet(trip_scen_sets)
```

Arguments

trip_scen_sets data frame of all trips from all scenarios

Value

total mMETs per week per person

trim_glm_object	<i>Reduce size of glm object</i>
-----------------	----------------------------------

Description

Delete some attributes of glm object in order to save space

Usage

```
trim_glm_object(obj)
```

Arguments

obj glm object

Value

glm object

`walk_to_pt_and_combine_scen`*Add walk to PT*

Description

Adds a short walk stage to any PT trip if required. Combines list of scenarios into one data frame

Usage

```
walk_to_pt_and_combine_scen(SYNTHETIC_TRIPS)
```

Arguments

`trip_set` list of data frames, trips from all scenarios

Value

data frame, all trips from all scenarios

Index

add_distance_columns, 1
add_ghost_trips, 2
add_trips, 2
add_walk_trips, 3
ap_dose_response_curve, 4
assign_age_groups, 4

beta_pointiness, 5

combine_health_and_pif, 6
combined_rr_ap_pa, 5
complete_trip_distance_duration, 6
create_all_scenarios, 7
create_cycle_scenarios, 7
create_max_mode_share_scenarios, 8
create_scenario, 8
create_synth_pop, 9
create_walk_scenario, 9

dirichlet_pointiness, 10
dist_dur_tbls, 11
distances_for_injury_function, 10

gen_ap_rr, 11
gen_pa_rr, 12
get_all_distances, 12
get_scenario_settings, 13
get_synthetic_from_trips, 13

health_burden, 14

injuries_function_2, 14
injury_death_to_yll, 15
ithim_calculation_sequence, 15
ithim_load_data, 16
ithim_setup_baseline_scenario, 16
ithim_setup_parameters, 17
ithim_uncertainty, 18

join_hb_and_injury, 19

PA_dose_response, 19
parallel_evppi_for_AP, 20
population_attributable_fraction, 20

run_ithim, 21
run_ithim_setup, 21

scale_trip_distances, 22
scenario_pm_calculations, 22
set_injury_contingency, 23
set_vehicle_inventory, 23
summarise_ithim_inputs, 23

total_mmet, 24
trim_glm_object, 24

walk_to_pt_and_combine_scen, 25