

Package: `treat.sim` (via `r-universe`)

September 14, 2024

Type Package

Title Nelson's Treatment Centre Simulation in Simmer

Version 0.3.0

Description A discrete-event simulation of a simple urgent care treatment centre simulation from Nelson (2013). Implemented in R Simmer. The model is packaged to allow for easy experimentation, summary of results, and implementation in other software such as a Shiny interface.

License MIT + file LICENSE

Encoding UTF-8

URL <https://github.com/pythonhealthdatascience/stars-treat-simmer>
<https://pythonhealthdatascience.github.io/stars-treat-simmer/>

BugReports <https://github.com/pythonhealthdatascience/stars-treat-simmer/issues>

LazyData true

RoxygenNote 7.3.2

Imports assertthat (\geq 0.2.1), dplyr (\geq 1.1.4), ggplot2 (\geq 3.5.1), simmer (\geq 4.4.6.4), simmer.bricks (\geq 0.2.2), RCurl (\geq 1.98-1.16), Rlab (\geq 4.0), magrittr (\geq 2.0.3), tidyselect (\geq 1.2.1), tidyr (\geq 1.3.1)

Depends R (\geq 4.1.1)

Suggests knitr, rmarkdown

VignetteBuilder knitr

Repository <https://pythonhealthdatascience.r-universe.dev>

RemoteUrl <https://github.com/pythonhealthdatascience/stars-treat-simmer>

RemoteRef HEAD

RemoteSha 161fa56392a836f1b3f6f32b448ae18ef29c9972

Contents

arrivals_by_replication	2
create_arrival_generator	3
create_experiment	3
create_non_trauma_pathway	5
create_nt_cubicle_treatment	6
create_summary_table	7
create_trauma_pathway	7
histogram_of_replications	8
multiple_replications	9
nelson_arrivals	9
normal_moments_from_lognormal	10
nspp_thinning	11
replication_results_table	12
resource_utilisation_by_replication	12
resource_waiting_times_by_replication	13
sample_arrival_type	13
sample_nt_trauma_treatment	14
single_run	15
system_kpi_for_rep_i	16
Index	17

arrivals_by_replication
Number of arrivals in each replication

Description

Returns the number of arrivals by replication as a data.frame

Usage

```
arrivals_by_replication(envs)
```

Arguments

envs simmer environments

Value

data.frame

`create_arrival_generator`

Create a trajectory to generate trauma and not-trauma arrivals to the treatment centre.

Description

Simulates the arrival of a patient, assigns a patient type and selects treatment trajectory.

Usage

```
create_arrival_generator(exp)
```

Arguments

`exp` An experiment in list form - contains all model parameters. Use `treat.sim::create_experiment()` to generate an experiment list.

Value

a simmer trajectory

See Also

[`create_experiment()`] to create list containing default and custom experimental parameters.

Examples

```
default_exp <- create_experiment()
patient_generator <- create_arrival_generator(default_exp)
```

`create_experiment`

Create a simulation experiment parameter list

Description

Create and return a list containing all of the parameters used by the `treat.sim` model.

Usage

```

create_experiment(
  n_triage_bays = DEFAULT_N_TRIAGE,
  n_reg_clerks = DEFAULT_N_REG,
  n_exam_rooms = DEFAULT_N_EXAM,
  n_trauma_rooms = DEFAULT_N_TRAUMA,
  n_non_trauma_cubicles = DEFAULT_NON_TRAUMA_CUBICLES,
  n_trauma_cubicles = DEFAULT_TRAUMA_CUBICLES,
  triage_mean = DEFAULT_TRIAGE_MEAN,
  stabilisation_mean = DEFAULT_TRAUMA_MEAN,
  trauma_treat_params = DEFAULT_TRAUMA_TREATMENT_PARAMS,
  reg_params = DEFAULT_REG_PARAMS,
  exam_params = DEFAULT_EXAM_PARAMS,
  prob_non_trauma_treat = DEFAULT_NON_TRAUMA_TREAT_P,
  nontrauma_treat_params = DEFAULT_NON_TRAUMA_TREATMENT_PARAMS,
  prob_trauma = DEFAULT_PROB_TRAUMA,
  arrival_profile = nelson_arrivals,
  log_level = LOG_LEVEL
)

```

Arguments

n_triage_bays	Number of triage bays
n_reg_clerks	Number of booking clerks
n_exam_rooms	Number of exam rooms
n_trauma_rooms	Number of trauma rooms for stabilisation
n_non_trauma_cubicles	Number of non-trauma treatment cubicles
n_trauma_cubicles	Number of trauma treatment cubicles
triage_mean	Mean triage duration (exponential distribution)
stabilisation_mean	Mean trauma stabilisation time (exponential distribution)
trauma_treat_params	list - mu and sigma for trauma treatment (lognormal)
reg_params	list - mu and sigma for registration (lognormal)
exam_params	list mu and sigma for examination time (normal)
prob_non_trauma_treat	probability trauma patient requires treatment
nontrauma_treat_params	list - mu and sigma for non trauma cubicle treatment (lognormal)
prob_trauma	probability arrival has trauma injuries
arrival_profile	time dependent arrival profile The default value used is the package internal dataset 'nelson_arrivals'. A new dataset should include three columns: 1. period (equally spaced 60 min intervals 0 to 1020) 2. arrival_rate (per hour) 3. arrival_rate2 (per minute)

log_level simmer log level. Set to 0 to hide all debug info as model runs.

Details

If no parameters are passed to the function then a default experiment is created.

Users can choose to create custom experiments by passing values to the corresponding parameters in the function.

Value

A list

Examples

```
# sample from Nelson arrivals at time 20.0
default_experiment <- create_experiment()

# set number of triage bays to 3
default_experiment <- create_experiment(n_triage_bays=3)
```

create_non_trauma_pathway

Create and return a trajectory for non-trauma patients

Description

Simulates the process for a non-trauma patient. Logic is:

1. Triage (requires triage bay)
2. Registration (requires clert)
3. Examination (requires exam room)
4. Probabilistic decision about treatment
- 4.1. Treatment (requires cubicle)
5. Discharge

Usage

```
create_non_trauma_pathway(exp)
```

Arguments

exp An experiment in list form - contains all model parameters. Use `treat.sim::create_experiment()` to generate an experiment list.

Value

a simmer trajectory

See Also

[create_experiment()] to create list containing default and custom experimental parameters.

Examples

```
default_exp <- create_experiment()
nt_traj <- create_non_trauma_pathway(default_exp)
```

create_nt_cubicle_treatment

Create and return a simmer::trajectory representing cubicle treatment of non-trauma patients.

Description

Simulates cubicle treatment of trauma patients (log normally distributed time)

Usage

```
create_nt_cubicle_treatment(exp)
```

Arguments

exp An experiment in list form - contains all model parameters. Use treat.sim::create_experiment() to generate an experiment list.

Value

a simmer trajectory

See Also

[create_experiment()] to create list containing default and custom experimental parameters.

Examples

```
default_exp <- create_experiment()
nt_treatment_traj <- create_nt_cubicle_treatment(default_exp)
```

create_summary_table *Create a summary table of multiple replications data*

Description

Accepts a table of replication results and returns the mean of those values in a data.frame

Usage

```
create_summary_table(rep_table, dp = 2)
```

Arguments

rep_table data.frame containing replications (rows) and KPIs (cols)
dp the number of decimal places

Value

data.frame

See Also

[replication_results_table()] to create a table of replications

create_trauma_pathway *Create and return a trauma pathway simmer trajectory*

Description

Trauma patients follow this process in the trajectory:

1. Triage (requires triage bay)
2. Stabilisation (requires trauma room)
3. Treatment (requirement trauma cubicle)
4. Discharge

Usage

```
create_trauma_pathway(exp)
```

Arguments

exp An experiment in list form - contains all model parameters. Use treat.sim::create_experiment() to generate an experiment list.

Value

a simmer trajectory

See Also

[create_experiment()] to create list containing default and custom experimental parameters.

Examples

```
default_exp <- create_experiment()
trauma_traj <- create_trauma_pathway(default_exp)
```

histogram_of_replications

Histogram of replications for a selected KPI

Description

Accepts a table of replication results and a ggplot histogram object for a selected column.

Usage

```
histogram_of_replications(rep_table, column_name, unit_label, n_bins = 10)
```

Arguments

rep_table	data.frame containing replications (rows) and KPIs (cols)
column_name	string name of the KPI to plot
unit_label	string of the x-axis label unit
n_bins	number of bins for the histogram

Value

plot

See Also

[replication_results_table()] to create a table of replications

multiple_replications *Perform multiple replications of the treat.sim model*

Description

The function runs `single_run` in a loop and returns the list of environments from each replication. The result can be used for analysis of the model

Usage

```
multiple_replications(exp, n_reps = 5, random_seed = 0)
```

Arguments

<code>exp</code>	An experiment in list form - contains all model parameters.
<code>n_reps</code>	number of replications to run.
<code>random_seed</code>	the random seed for the reps

Value

a list of simmer environments

See Also

[`single_run()`] to perform a single replication with the model

Examples

```
exp <- create_experiment(log_level=0)

# run 50 replications of the model
reps <- multiple_replications(exp, n_reps=50, random_seed=0)
```

nelson_arrivals *Time dependent arrival profile from Nelson (2013)*

Description

A simple default arrival profile for use with `treat.sim`

Usage

```
nelson_arrivals
```

Format

```
## 'nelson_arrivals' This is a data.frame containing the following columns
```

```
period 0 to 480 in 60 minute intervals
```

```
arrival_rate rate per hour
```

```
arrival_rate2 rate per minute
```

Source

```
'https://raw.githubusercontent.com/TomMonks/open-science-for-sim/main/src/notebooks/01_foss_sim/data/ed_arrivals.csv'
```

```
normal_moments_from_lognormal
```

```
Mean and Variance of the underlying Normal Distribution
```

Description

Calculates the mu and sigma of the normal distribution underlying a lognormal

Usage

```
normal_moments_from_lognormal(mean, std)
```

Arguments

```
mean          A number. Sample mean.
```

```
std           A number. Sample standard deviation
```

Details

'rlnorm' from 'stats' is designed to sample from the lognormal distribution. The parameters it expects are moments of the underlying normal distribution. Using sample mean and standard deviation this function calculates the mu and sigma of the normal distribution.

Value

A list containing mu and sigma

Source

```
https://blogs.sas.com/content/iml/2014/06/04/simulate-lognormal-data-with-specified-mean-and-variance.html
```

Examples

```
normal_moments_from_lognormal(mean = 125.0, std = 5.0)
```

nspp_thinning	<i>Non-stationary Poisson Process (NSPP) via thinning</i>
---------------	---

Description

Use the current simulation time to sample an appropriate inter-arrival time from a user set NSPP. Assumes equal spaced intervals.

Usage

```
nspp_thinning(simulation_time, arrival_profile, debug = FALSE)
```

Arguments

simulation_time	A number. The current simulation time
arrival_profile	A data.frame The time dependent arrival profile
debug	bool. TRUE = printout log of thinning after a sample has been accepted. FALSE = no debug info provided.

Details

Thinning is an acceptance-rejection approach to sampling inter-arrival times (IAT) from a time dependent distribution where each time period follows its own exponential distribution.

There are two random variables employed in sampling: an exponential distribution (used to sample IAT) and a uniform distribution (used to accept/reject samples).

All IATs are sampled from an Exponential distribution with the highest arrival rate (most frequent). These arrivals are then rejected (thinned) proportional to the ratio of the current arrival rate to the maximum arrival rate. The algorithm executes until a sample is accepted. The IAT returned is the sum of all the IATs that were sampled.

Value

A number

Examples

```
# sample from Nelson arrivals at time 20.0  
nspp_thinning(20.0, nelson_arrivals, debug=TRUE)
```

`replication_results_table`*Function to create the replications table*

Description

Accepts a list of simmer environments and converts to a data.frame of replications (rows) x KPIs (cols).

Usage

```
replication_results_table(  
  reps,  
  exp,  
  results_collection_period = DEFAULT_RESULTS_COLLECTION_PERIOD  
)
```

Arguments

`reps` list of simmer environments
`exp` list of "experiment" contains all of the model parameters used to create the results
`results_collection_period` the length of time results were collected.

Value

data.frame

`resource_utilisation_by_replication`*Resource utilisation by replication*

Description

Returns the utilisation of resource by replication as a data.frame calculation: total busy time / total scheduled resource time. where total scheduled time = `n_resource` * results collection period.

Usage

```
resource_utilisation_by_replication(reps, exp, results_collection_period)
```

Arguments

exp list of parameters for experiment
 results_collection_period results collection simulation time
 envs simmer environments

Value

data.frame

resource_waiting_times_by_replication
Waiting times for resources by replication

Description

Returns the mean waiting times for resource by replication as a data.frame

Usage

resource_waiting_times_by_replication(reps)

Arguments

envs simmer environments from each replication of the model

Value

data.frame

sample_arrival_type *Sample a patient type*

Description

'sample_arrival_type' samples if a patient type is trauma or non-trauma with a given probability.

Usage

sample_arrival_type(p, n = 1)

Arguments

p A number: the probability a patient has trauma on arrival
 n A number: number of samples to generate.

Details

The function uses the Bernoulli distribution (Rlab) to sample if a patient is Trauma or Non-Trauma. The return values are 1 = Trauma, 2 = Non-trauma.

Examples

```
patient_type = sample_arrival_type(0.4)
```

```
sample_nt_trauma_treatment
```

Sample a if a non-trauma patient requires treatment

Description

‘sample_nt_trauma_treatment’ samples if a non-trauma patient requires cubicle treatment

Usage

```
sample_nt_trauma_treatment(p)
```

Arguments

p A number: The probability the patient requires treatment

Details

The function uses the Bernoulli distribution (Rlab) to sample if a patient is requires treatment or not. The return values are 1 = Treatment, 0 = No treatment

Examples

```
treat_flag = sample_nt_trauma_treatment(0.30)
```

single_run	<i>Perform a single replication of the model</i>
------------	--

Description

The function adds treat.sim resources to an environment. The model is then run for a single replication. The environment is returned for use in results analysis.

Usage

```
single_run(  
  exp,  
  rep_number = 1,  
  run_length = DEFAULT_RESULTS_COLLECTION_PERIOD,  
  debug_arrivals = FALSE  
)
```

Arguments

exp	An experiment in list form - contains all model parameters.
rep_number	the replication number (default=1)
run_length	run length of the simulation (default=1020 minutes)
debug_arrivals	boolean flag to show debug info for the thinning process Use treat.sim::create_experiment() to generate an experiment list.

Value

a simmer environment with a completed model run.

See Also

[create_experiment()] to create list containing default and custom experimental parameters.

Examples

```
set.seed(42)  
exp <- create_experiment(log_level=0)  
treat_sim <- single_run(exp)  
print("Simulation Complete.")
```

system_kpi_for_rep_i *System level KPIs*

Description

Calculates mean time in system and throughput for a replication

Usage

```
system_kpi_for_rep_i(reps, rep_i)
```

Arguments

reps	list of simmer environments
rep_i	replication for calculation

Value

data.frame

Index

* datasets

- nelson_arrivals, 9
- arrivals_by_replication, 2
- create_arrival_generator, 3
- create_experiment, 3
- create_non_trauma_pathway, 5
- create_nt_cubicle_treatment, 6
- create_summary_table, 7
- create_trauma_pathway, 7
- histogram_of_replications, 8
- multiple_replications, 9
- nelson_arrivals, 9
- normal_moments_from_lognormal, 10
- nspp_thinning, 11
- replication_results_table, 12
- resource_utilisation_by_replication, 12
- resource_waiting_times_by_replication, 13
- sample_arrival_type, 13
- sample_nt_trauma_treatment, 14
- single_run, 15
- system_kpi_for_rep_i, 16