Package ‘NHSRwaitinglist’

July 15, 2025

Title Waiting List Metrics Using Queuing Theory

Version 0.1.2

Maintainer Chris Mainey <c.mainey1@nhs.net>

Description Waiting list management using queuing theory to analyse, predict and manage queues, based on the approach described in Fong et al. (2022) <doi:10.1101/2022.08.23.22279117>. Aimed at UK National Health Service (NHS) applications, waiting list summary statistics, target-value calculations, waiting list simulation, and scheduling functions are included.

License MIT + file LICENSE

Encoding UTF-8

RoxygenNote 7.3.2

VignetteBuilder knitr

URL https://nhs-r-community.github.io/NHSRwaitinglist/,
https://github.com/nhs-r-community/NHSRwaitinglist

BugReports https://github.com/nhs-r-community/NHSRwaitinglist/issues

Config/testthat/edition 3

Imports cli, dplyr, rlang, utils, stats, randomNames

Suggests ggplot2, knitr, rmarkdown, testthat (>= 3.0.0)

Depends R (>= 4.1.0)

LazyData true

Language en-GB

NeedsCompilation no

Author Neil Walton [aut] (ORCID: <https://orcid.org/0000-0002-5241-9765>),
Jacqueline Grout [ctb],
Zoë Turner [ctb] (ORCID: <https://orcid.org/0000-0003-1033-9158>),
Matt Dray [aut],
Paul Fenton [ctb],
Peter Shakeshaft [ctb],
David Foord [ctb],
Tom Smith [aut],

Chris Mainey [cre, aut] (ORCID:
<https://orcid.org/0000-0002-3018-6171>),
Mohammed Mohammed [ctb],
NHS-R community [cph]

Repository CRAN

Date/Publication 2025-07-15 15:50:02 UTC

Contents

`calc_queue_load` *Calculate Queue Load*

Description

Calculates the queue load. The queue load is the number of arrivals that occur for every patient leaving the queue (given that the waiting list did not empty). It could also be described as the rate of service at the queue. The queue load is calculated by dividing the demand by the capacity: queue_load = demand / capacity.

Usage

```
calc_queue_load(demand, capacity)
```

Arguments

`demand` Numeric value of rate of demand in same units as target wait - e.g. if target wait is weeks, then demand in units of patients/week.

`capacity` Numeric value of the number of patients that can be served (removals) from the waiting list each week.

Value

Numeric value of load which is the ratio between demand and capacity.

Examples

```
# If 30 patients are added to the waiting list each week (demand) and 27
# removed (capacity) this results in a queue load of 1.11 (30/27).
calc_queue_load(30, 27)
```

`calc_relief_capacity` *Relief Capacity*

Description

Calculates required relief capacity to achieve target queue size in a given period of time as a function of demand, queue size, target queue size and time period. Relief Capacity is required if Queue Size > 2 * Target Queue Size.

Relief Capacity = Current Demand + (Queue Size - Target Queue Size)/Time Steps

WARNING!: make sure units match. I.e. if demand is measured per week then time_to_target should be weeks or if demand is per day then time_to_target is per day

Usage

```
calc_relief_capacity(
  demand,
  queue_size,
  target_queue_size,
  time_to_target = 26,
  num_referrals = 0,
  cv_demand = 0
)
```

Arguments

`demand` Numeric value of rate of demand in same units as target wait e.g. if target wait is weeks, then demand in units of patients/week.

`queue_size` Numeric value of current number of patients in queue.

`target_queue_size`
Numeric value of desired number of patients in queue.

`time_to_target` Numeric value of desired number of time-steps to reach the target queue size by.

`num_referrals` Numeric value of the number of referrals per time step.

`cv_demand` To be completed

Value

A numeric value of the required rate of capacity to achieve a target queue size in a given period of time.

Examples

```
# If demand is 30 patients per week, the current queue size is 1200 and the
# target is to achieve a queue size of 390 in 26 weeks, then

# Relief Capacity = 30 + (1200 - 390)/26 = 61.15 patients per week.

calc_relief_capacity(30, 1200, 390, 26)
```

`calc_target_capacity` *Target Capacity*

Description

Applies Kingman/Marchal's Formula :

```
 capacity = demand + (cvd**2 + cvc**2) / waiting_time
```

where cvd = coefficient of variation of time between arrivals cvd = coefficient of variation of service times waiting_time = target_wait / factor

Usage

```
calc_target_capacity(
  demand,
  target_wait,
  factor = 4,
  cv_demand = 1,
  cv_capacity = 1
)
```

Arguments

`demand`	Numeric value of rate of demand in same units as target wait e.g. if target wait is weeks, then demand in units of patients/week.
`target_wait`	Numeric value of number of weeks that has been set as the target within which the patient should be seen.
`factor`	the amount we divide the target by in the waiting list e.g. if target is 52 weeks the mean wait should be 13 for a factor of 4
`cv_demand`	coefficient of variation of time between arrivals
`cv_capacity`	coefficient of variation between removals due to operations completed

Value

numeric. The capacity required to achieve a target waiting time.

Examples

```
demand <- 4 # weeks
target_wait <- 52 # weeks

# number of operations per week to have mean wait of 52/4
calc_target_capacity(demand, target_wait)
```

`calc_target_mean_wait` *Average Waiting Time*

Description

This calculates the target mean wait given the two inputs of target_wait and a numerical value for factor. The average wait is actually the target mean wait and is calculated as follows: target_wait / factor. If we want to have a chance between 1.8%-0.2% of making a waiting time target, then the average patient should have a waiting time between a quarter and a sixth of the target. Therefore: The mean wait should sit somewhere between target_wait/factor=6 < Average Waiting Time < target_wait/factor=4.

Usage

```
calc_target_mean_wait(target_wait, factor = 4)
```

Arguments

`target_wait`	Numeric value of the number of weeks that has been set as the target within which the patient should be seen.
`factor`	Numeric factor used in average wait calculation - to get a quarter of the target use factor=4 and one sixth of the target use factor = 6 etc. Defaults to 4.

Value

Numeric value of target mean waiting time to achieve a given target wait.

Examples

```
# If the target wait is 52 weeks then the target mean wait with a factor of 4
# would be 13 weeks and with a factor of 6 it would be 8.67 weeks.
calc_target_mean_wait(52, 4)
```

calc_target_queue_size

Target Queue Size

Description

Uses Little's Law to calculate the target queue size to achieve a target waiting time as a function of observed demand, target wait and a variability factor used in the target mean waiting time calculation.

Target Queue Size = Demand * Target Wait / 4.

The average wait should sit somewhere between target_wait/factor=6 < Average Waiting Time < target_wait/factor=4 The factor defaults to 4.

Only applicable when Capacity > Demand.

Usage

```
calc_target_queue_size(demand, target_wait, factor = 4)
```

Arguments

`demand`	Numeric value of rate of demand in same units as target wait e.g. if target wait is weeks, then demand in units of patients/week.
`target_wait`	Numeric value of number of weeks that has been set as the target within which the patient should be seen.
`factor`	Numeric factor used in average wait calculation • to get a quarter of the target use factor=4 • to get one sixth of the target use factor = 6 etc. Defaults to 4.

Value

Numeric target queue length.

Examples

```
# If demand is 30 patients per week and the target wait is 52 weeks, then the
# Target queue size = 30 * 52/4 = 390 patients.

calc_target_queue_size(30, 52, 4)
```

calc_waiting_list_pressure *Calculate Waiting List Pressure*

Description

For a waiting list with target waiting time, the pressure on the waiting list is twice the mean delay divided by the waiting list target. The pressure of any given waiting list should be less than 1. If the pressure is greater than 1 then the waiting list is most likely going to miss its target. The waiting list pressure is calculated as follows: pressure = 2 * mean_wait / target_wait.

Usage

```
calc_waiting_list_pressure(mean_wait, target_wait)
```

Arguments

mean_wait	Numeric value of target mean waiting time to achieve a given target wait.
target_wait	Numeric value of the number of weeks that has been set as the target within which the patient should be seen.

Value

Numeric value of wait_pressure which is the waiting list pressure.

Examples

```
calc_waiting_list_pressure(63, 52)
```

create_waiting_list *Create Waiting List*

Description

Creates a waiting list using the parameters specified

Usage

```
create_waiting_list(
  n,
  mean_arrival_rate,
  mean_wait,
  start_date = Sys.Date(),
  limit_removals = TRUE,
  sd = 0,
  rott = 0,
  ...
)
```

Arguments

`n` Numeric value of rate of demand in same units as target wait

- e.g. if target wait is weeks, then demand in units of patients/week.

`mean_arrival_rate` Numeric value of mean daily arrival rate.

`mean_wait` Numeric value of mean wait time for treatment/on waiting list.

`start_date` Character value of date from which to start generated waiting list.

`limit_removals` Defaults to TRUE

`sd` Numeric value, standard deviation. Defaults to 0.

`rott` Numeric value, proportion of referrals to be randomly flagged as ROTT. Defaults to 0.

`...` Container for the list

Value

A tibble with randomly generated patient records and the following columns:

pat_id Integer. Unique identifier for the patient.

addition_date Date. The date the patient was added to the waiting list.

removal_date Date. The date the patient was removed from the waiting list.

wait_length Numeric. Number of days between the addition and removal dates.

rott Logical. Whether the removal was for reasons other than treatment (ROTT).

Additional columns may be included if supplied via `...`, where named vectors (e.g., patient-level variables) of compatible length are merged into the output tibble.

Examples

```
create_waiting_list(366, 50, 21, "2024-01-01", 10, 0.1)
```

`demographic_data` *demographic data*

Description

demographic data

Usage

```
data(demographic_data)
```

Format

Data frame with 9 columns

hospital_site ODS hospital site code

... Others to do with file is updated

Examples

```
data(demographic_data)
```

opcs4 *OPCS4 data*

Description

OPCS4 data

Usage

```
data(opcs4)
```

Format

Data frame with 9 columns

code_1digit The first digit of the OPCS4 code, or ’chapter’

name_1digit The name/group of ’chapter’ of the OPCS4 code

... Others to do with file is updated

Source

https://biobank.ndph.ox.ac.uk/ukb/coding.cgi?id=240

Examples

```
data(opcs4)
```

sim_patients *Generator of NHS patients*

Description

Generates simulated NHS patients

Usage

```
sim_patients(n_rows = 10, start_date = NULL)
```

Arguments

n_rows	Number of rows/patients to generate
start_date	Start date (needed to generate patient ages)

Value

A data.frame representing an empty waiting list with the following columns:

Referral Date. Referral date; all values are `NA`.

Removal Date. Removal date; all values are `NA`.

Withdrawal Date. Patient withdrawal date; all values are `NA`

Priority Numeric. Waiting list priority level, from 1 (most urgent) to 4 (least urgent).

Target_wait Numeric. Target number of days the patient should wait at the assigned priority level (e.g., 28 days for priority 2)

Name Character. Patient name in the format `"Last, First"`.

Birth_date Date. Date of birth.

NHS_number Integer. Patient identifier, up to 100,000,000.

Specialty_code Character. One-letter code representing the specialty of the procedure.

Specialty Character. Full name of the specialty associated with the procedure.

OPCS Character. OPCS-4 code of the selected procedure.

Procedure Character. Name of the selected procedure.

Consultant Character. Consultant name in the format `"Last, First"`.

Examples

```
sim_patients()
```

`sim_schedule` *Generator a list of dates to schedule*

Description

Generates a list if dates in a given range

Usage

```
sim_schedule(n_rows = 10, start_date = NULL, daily_capacity = 1)
```

Arguments

`n_rows` Number of rows/patients to generate

`start_date` Start date (needed to generate patient ages)

`daily_capacity` Number of patients per day

Value

A vector of `Date` values representing scheduled procedure dates. The length of the vector is equal to `n_rows`, and the dates are spaced according to the specified `daily_capacity`.

`wl_insert` *Insert new referrals into the waiting list*

Description

Adds new referrals, with other columns set as `NA`.

Usage

```
wl_insert(waiting_list, additions, referral_index = 1)
```

Arguments

`waiting_list` data.frame. A df of referral dates and removals

`additions` Date or character vector (in format 'YYYY-MM-DD'). A list of referral dates to add to the waiting list

`referral_index` The index of the column in `waiting_list` which contains the referral dates. Defaults to the first column.

Value

A `data.frame` representing the updated waiting list, with additional referrals dates in the column specified by `referral_index`. Other columns are filled with `NA` in the new rows. The result is sorted by the referral column.

Examples

```
referrals <- c.Date("2024-01-01", "2024-01-04", "2024-01-10", "2024-01-16")
removals <- c.Date("2024-01-08", NA, NA, NA)
waiting_list <- data.frame("referral" = referrals, "removal" = removals)
additions <- c.Date("2024-01-03", "2024-01-05", "2024-01-18")
longer_waiting_list <- wl_insert(waiting_list, additions)
```

wl_join *Join two waiting list*

Description

Take two waiting list and sorting in date order

Usage

```
wl_join(wl_1, wl_2, referral_index = 1)
```

Arguments

`wl_1`	a waiting list: dataframe consisting addition and removal dates
`wl_2`	a waiting list: dataframe consisting addition and removal dates
`referral_index`	the column index where referrals are listed

Value

A data.frame representing the combined waiting list, created by joining `wl_1` and `wl_2`. The result is sorted by the referral date column specified by `referral_index`. The column structure is preserved from the input data frames.

Examples

```
referrals <- c.Date("2024-01-01","2024-01-04","2024-01-10","2024-01-16")
removals <- c.Date("2024-01-08",NA,NA,NA)
wl_1 <- data.frame("referral" = referrals ,"removal" = removals )

referrals <- c.Date("2024-01-04","2024-01-05","2024-01-16","2024-01-25")
removals <- c.Date("2024-01-09",NA,"2024-01-19",NA)
wl_2 <- data.frame("referral" = referrals ,"removal" = removals )
wl_join(wl_1,wl_2)
```

`wl_queue_size` *Queue size calculator*

Description

Calculates queue sizes from a waiting list

Usage

```
wl_queue_size(
  waiting_list,
  start_date = NULL,
  end_date = NULL,
  referral_index = 1,
  removal_index = 2
)
```

Arguments

`waiting_list`	data.frame consisting addition and removal dates
`start_date`	Date or character (in format 'YYYY-MM-DD'); start of calculation period
`end_date`	Date or character (in format 'YYYY-MM-DD'); end of calculation period
`referral_index`	the index of referrals in waiting_list
`removal_index`	the index of removals in waiting_list

Value

A data.frame containing the size of the waiting list for each day in the specified date range. If `start_date` and/or `end_date` are `NULL`, the function uses the earliest and latest referral dates in the input data.frame. The returned data.frame has the following columns:

dates Date. Each date within the computed range, starting from the first referral.

queue_size Numeric. Number of patients on the waiting list on that date.

Examples

```
referrals <- c.Date("2024-01-01", "2024-01-04", "2024-01-10", "2024-01-16")
removals <- c.Date("2024-01-08", NA, NA, NA)
waiting_list <- data.frame("referral" = referrals, "removal" = removals)
wl_queue_size(waiting_list)
```

`wl_referral_stats` *Calculate some stats about referrals*

Description

Calculate some stats about referrals

Usage

```
wl_referral_stats(
  waiting_list,
  start_date = NULL,
  end_date = NULL,
  referral_index = 1
)
```

Arguments

`waiting_list`	data.frame. A df of referral dates and removals
`start_date`	Date or character (in format 'YYYY-MM-DD'); The start date to calculate from
`end_date`	Date or character (in format 'YYYY-MM-DD'); The end date to calculate to
`referral_index`	the column index of referrals

Value

A data.frame with the following summary statistics on referrals/demand:

demand_weekly Numeric. Mean number of additions to the waiting list per week.

demand_daily Numeric. Mean number of additions to the waiting list per day.

demand_cov Numeric. Coefficient of variation in the time between additions to the waiting list.

demand_count Numeric. Total demand over the full time period.

Examples

```
referrals <- c.Date("2024-01-01", "2024-01-04", "2024-01-10", "2024-01-16")
removals <- c.Date("2024-01-08", NA, NA, NA)
waiting_list <- data.frame("referral" = referrals, "removal" = removals)
referral_stats <- wl_referral_stats(waiting_list)
```

wl_removal_stats *Calculate some stats about removals*

Description

Calculate some stats about removals

Usage

```
wl_removal_stats(
  waiting_list,
  start_date = NULL,
  end_date = NULL,
  referral_index = 1,
  removal_index = 2
)
```

Arguments

`waiting_list`	data.frame. A df of referral dates and removals
`start_date`	Date or character (in format 'YYYY-MM-DD'); The start date to calculate from.
`end_date`	Date or character (in format 'YYYY-MM-DD'); The end date to calculate to.
`referral_index`	Index of the referral column in waiting_list.
`removal_index`	Index of the removal column in waiting_list.

Value

A data.frame with the following summary statistics on removals/capacity:

capacity_weekly Numeric. Mean number of removals from the waiting list per week.

capacity_daily Numeric. Mean number of removals from the waiting list per day.

capacity_cov Numeric. Coefficient of variation in the time between removals from the waiting list.

removal_count Numeric. Total number of removals from the waiting list over the full time period.

Examples

```
referrals <- c.Date("2024-01-01", "2024-01-04", "2024-01-10", "2024-01-16")
removals <- c.Date("2024-01-08", NA, NA, NA)
waiting_list <- data.frame("referral" = referrals, "removal" = removals)
removal_stats <- wl_removal_stats(waiting_list)
```

wl_schedule *A simple operation scheduler*

Description

Takes a list of dates and schedules them to a waiting list, by adding a removal date to the data.frame. This is done in referral date order, I.e. earlier referrals are scheduled first (FIFO).

Usage

```
wl_schedule(
  waiting_list,
  schedule,
  referral_index = 1,
  removal_index = 2,
  unscheduled = FALSE
)
```

Arguments

`waiting_list`	data.frame. A df of referral dates and removals
`schedule`	Date or character vector. Should be formatted as year-month-date, e.g. "2024-04-01". The dates to schedule open referrals into (i.e. dates of unbooked future capacity)
`referral_index`	The column index in the waiting_list which contains the referral dates
`removal_index`	The column index in the waiting_list which contains the removal dates
`unscheduled`	logical. If TRUE, returns a list of scheduled and unscheduled procedures If FALSE, only returns the updated waiting list

Value

The updated waiting list with removal dates assigned based on the given schedule, either as a single `data.frame` (default) or as part of a list (if `unscheduled` = `TRUE`).

If `unscheduled` = `TRUE`, returns a `list` with two data frames:

1. A `data.frame`. The updated waiting list with scheduled removals.
2. A `data.frame` showing which slots were used, with columns:

   schedule Date. The available dates from the input `schedule`.

   scheduled Numeric. `1` if the slot was used to schedule a patient, `0` if not.

Examples

```
referrals <- c.Date("2024-01-01", "2024-01-04", "2024-01-10", "2024-01-16")
removals <- c.Date("2024-01-08", NA, NA, NA)
waiting_list <- data.frame("referral" = referrals, "removal" = removals)
schedule <- c.Date("2024-01-03", "2024-01-05", "2024-01-18")
updated_waiting_list <- wl_schedule(waiting_list, schedule)
```

`wl_simulator` *Simple simulator to create a waiting list*

Description

Creates a simulated waiting list comprising referral dates, and removal dates

Usage

```
wl_simulator(
  start_date = NULL,
  end_date = NULL,
  demand = 10,
  capacity = 11,
  waiting_list = NULL,
  withdrawal_prob = NA_real_,
  detailed_sim = FALSE
)
```

Arguments

`start_date`	Date or character (in format 'YYYY-MM-DD'); The start date to calculate from
`end_date`	Date or character (in format 'YYYY-MM-DD'); The end date to calculate to
`demand`	numeric. Weekly demand (i.e., typical referrals per week).
`capacity`	numeric. Weekly capacity (i.e., typical removals per week).
`waiting_list`	data.frame. Waiting list where each row is a pathway/patient with date columns 'Referral' and 'Removal'.
`withdrawal_prob`	numeric. Probability of a patient withdrawing.
`detailed_sim`	logical. If TRUE, simulation provides detailed output.

Value

A `data.frame` simulating a waiting list, with columns:

`Referral`	Date. The date each patient was added to the waiting list.

`Removal` Date. The date each patient was removed from the waiting list (may be `NA` if unscheduled).

If `detailed_sim = TRUE`, returns a more detailed `data.frame` with the following additional fields:

`Withdrawal` Date. The date the patient withdrew from the waiting list.

`Priority` Numeric. Waiting list priority level, from 1 (most urgent) to 4 (least urgent).

`Target_wait` Numeric. Target number of days the patient should wait at the assigned priority level (e.g., 28 days for priority 2)

`Name` Character. Patient name in the format `"Last, First"`.

`Birth_date` Date. Date of birth.

`NHS_number` Integer. Patient identifier, up to 100,000,000.

`Specialty_code` Character. One-letter code representing the specialty of the procedure.

`Specialty` Character. Full name of the specialty associated with the procedure.

`OPCS` Character. OPCS-4 code of the selected procedure.

`Procedure` Character. Name of the selected procedure.

`Consultant` Character. Consultant name in the format `"Last, First"`.

Examples

```
over_capacity_simulation <-
  wl_simulator("2024-01-01", "2024-03-31", 100, 110)
under_capacity_simulation <-
  wl_simulator("2024-01-01", "2024-03-31", 100, 90)
```

`wl_stats` *Calculate some stats about the waiting list*

Description

A summary of all the key stats associated with a waiting list

Usage

```
wl_stats(waiting_list, target_wait = 4, start_date = NULL, end_date = NULL)
```

Arguments

`waiting_list` data.frame. A df of referral dates and removals

`target_wait` numeric. The required waiting time

`start_date` Date or character (in format 'YYYY-MM-DD'); The start date to calculate from

`end_date` Date or character (in format 'YYYY-MM-DD'); The end date to calculate to

Value

A data.frame of key waiting list summary statistics based on queueing theory:

mean_demand Numeric. Mean number of additions to the waiting list per week.

mean_capacity Numeric. Mean number of removals from the waiting list per week.

load Numeric. Ratio between demand and capacity.

load_too_big Logical. Whether the load is greater than or equal to 1, indicating whether the waiting list is unstable and expected to grow.

count_demand Numeric. Total demand (i.e., number of referrals) over the full time period.

queue_size Numeric. Number of patients on the waiting list at the end of the time period.

target_queue_size Numeric. The recommended size of the waiting list to achieve approximately 98.2% of patients being treated within their target wait time. This is based on Little's Law, assuming the system is in equilibrium, with the average waiting time set to one-quarter of the `target_wait`.

queue_too_big Logical. Whether `queue_size` is more than twice the `target_queue_size`. A value of `TRUE` indicates the queue is at risk of missing its targets.

mean_wait Numeric. Mean waiting time in weeks.

cv_arrival Numeric. Coefficient of variation in the time between additions to the waiting list.

cv_removal Numeric. Coefficient of variation in the time between removals from the waiting list.

target_capacity Numeric. The weekly treatment capacity required to maintain the waiting list at its target equilibrium, assuming the target queue size has been reached.

relief_capacity Numeric. The temporary weekly capacity required to reduce the waiting list to its `target_queue_size` within 26 weeks, assuming current demand remains steady. Calculated only if `queue_too_big` is `TRUE`; otherwise returns `NA`.

pressure Numeric. A measure of pressure on the system, defined as `2 × mean_wait / target_wait`. Values greater than 1 suggest the system is unlikely to meet its waiting time targets.

Examples

```
referrals <- c.Date("2024-01-01", "2024-01-04", "2024-01-10", "2024-01-16")
removals <- c.Date("2024-01-08", NA, NA, NA)
waiting_list <- data.frame("referral" = referrals, "removal" = removals)
waiting_list_stats <- wl_stats(waiting_list)
```

Index