readme

This document contains an overview of all the files in the research and model.

**Table of content**

[Assign Surgery 1](#_Toc170142297)

[Assign surgery Empirical 2](#_Toc170142298)

[Assigning wards 2](#_Toc170142299)

[ICK Assignment 2](#_Toc170142300)

[Model 2](#_Toc170142301)

[Run Me 3](#_Toc170142302)

[Set resrouce Capacity 4](#_Toc170142303)

[Visualisation 4](#_Toc170142304)

[Ward OT scheduling 5](#_Toc170142305)

[Input variable creation 5](#_Toc170142306)

[Assigning group numbers 5](#_Toc170142307)

[Distribuitons length of stay 6](#_Toc170142308)

[Distribuitons surgery duration 6](#_Toc170142309)

[functions distributions 6](#_Toc170142310)

[probabilities 7](#_Toc170142311)

[total dataset 7](#_Toc170142312)

[Input data 7](#_Toc170142313)

[Model data 7](#_Toc170142314)

[results 7](#_Toc170142315)

# Assign Surgery

Contains three functions. The primary function is ‘Assign\_surgery’ which contains the process of taking the schedule information and assigning all the information to initiate the patient entity. Uses the functions ‘get\_surgery\_duration’ and ‘get\_ward\_stay’ (which are also in the file) to get the necessary data. These two functions use the fitted distributions as input to draw a corresponding value. All the information is stored in a list. The average duration of surgery and cleaning time is also used to determine an estimated ending time which can be used for the next patient.

**Packages:**

* Numpy
* Scipy.stats

**Returns:** Estimated ending time (int) and patient info (list)

# Assign surgery Empirical

Similar to Assing Surgery, however, it uses the testing data directly from the data frame instead of the distributions. This function is not used in the standard version of the model. It is only used when wanting to use the real data as input.

**Packages:**

* pandas
* random

**Returns:** Estimated ending time (int) and patient info (list)

# Assigning wards

This file contains one function, ‘assigning\_ward’; the goal is to define which ward a patient or entity should be assigned. It uses the information of the environment to collect the current availability on each of the wards. It uses the average length of stay and arrival time of the entity requesting the ward to determine whether the entity can be assigned a daycare unit. It uses the category to determine which MCU fits when the daycare unit cannot be assigned. When the best fitting MCU is full, there is always a backup; this backup, if this one is also full, the function returns both so the patient can wait until either becomes available.

It can also be used to assign an MCU when initially a daycare was assigned, but the daycare had to close, and the patient had to remain. For this, a factor is built in that can indicate the daycare option being skipped.

**Returns:** Ward (String) or Wards (list)

# ICK Assignment

This file contains one function, ‘Assigning\_ICK’; the goal is to define which ICK the patient needs to be assigned. This is based on the department of the surgery.

**Returns:** ICK (String)

# Model

This file contains the entire flow of the model containing one function which is used to call the model and three classes that are part of the model system:

**run\_simualtion:** It uses the model parameters and the input data frames to set up and run the model. First, it set up the required environment, counting and data frames. Uses the model's input parameters created to change the model's parameters. Sets up the resources like wards and Operating theatres. Creates lists to store the information. And then runs the model for the designated time. When the model is done running, it converts the lists to data frames and returns the data frames.

**TimestepAction (class):** This class periodically updates and collects resource usage data. It initialises resource data collection and sets the resource capacity at the start and regular intervals. The class maintains a loop that updates the resource capacities and collects data on the available and claimed quantities. This data is stored in a list.

**DaystepAction (class):** This class manages the daily scheduling. It retrieves the relevant surgery schedule for the day in the 28-day cycle. For each open OT, the class assigns patients to a slot and retrieves all the needed information; it initiates the entity. The class is initiated at the beginning of every day, initiating the patients for that day and then holding for 1440 time steps, to start again the next day.

**Patients (class):** This class represents individual patients undergoing surgeries. It sets up each patient's details, including their surgery schedule, ward stay, and probability of needing ICK care. During the simulation, the class handles the patient's arrival, surgery, ICK stay, and ward stay, ensuring proper resource allocation and capacity management. The class also manages potential delays, checks for OT overtime, and handles the transition between wards and ICK as needed. Finally, it records successful surgeries and ensures patients are discharged at appropriate times while keeping track of ICK occupancy and ward capacities.

**Returns:**

* Data frame with surgeries cancelled due to ward unavailability
* Data frame with surgeries cancelled due to OT unavailability
* Data frame with successful surgeries
* Data frame with the capacity use of the wards
* Data frame with occurrences of the OT going into overtime
* Count of the number of surgeries the system has tried to schedule
* Count of the number of times the wards had to over capacitate

**Packages:**

* Salabim
* Pandas
* Numpy
* Random

**Uses other files:**

* Ward\_OT\_scheduling
* Set\_resource\_capacity
* Assigning\_wards
* ICK\_assginment
* Assign\_Surgery

# Run Me

This file calls upon the ‘run\_simulation’ function in the model. To correctly run the function, the function ‘running’ is set. This function takes several inputs:

* The number of runs
* The length of the run, the number of repeats of the schedule
* The file path in which the results should be saved
* The schedule that the system should use
* Parameters to adjust model input, including:
  + Lower\_ward\_capacity: a factor to multiply the ward capacity for each ward
  + Increase\_surgery\_duration: a factor to multiply the length of surgery with
  + Increase\_ward\_stay: a factor to multiply the length of stay with
  + Increase\_IC\_probability: a factor to multiply the probability of the ICK assignment with (in the model, this is maximised at 100%, so a multiplication of 10 does not result in 10 times as many patients being assigned an ICK bed

The model then uses the Visualisation file to setup up the different data frames and plot in the designated file. It also uses the information returned by a model to set up a data frame with a count of all the output. Additionally, the model is set up to run multiple runs in parallel.

**Packages:**

* Pandas
* os
* matplotlib.pyplot
* Concurrent.futures

**Uses other files:**

* Model
* Visualisation

**Returns:** Counts data frame

# Set resource Capacity

This file contains one function, ‘set\_resoruce\_capacity’and used to use the environment schedule and current time and set the capacity for all wards and OT to the value the schedule designates. This is called by the time step function and is called once an hour (default) and updates the capacity.

# Visualisation

This file contains four functions to create the visualisation using the model's output:

**Ward\_utilisation:** This function calculates and visualises the utilisation of the wards based on resource use data. It combines the resource data across multiple simulation runs and calculates each day's utilisation. The function then creates boxplots per scheduled day for each ward and for the hospital in total. It then saves the plot to the designated directory.

**OT\_utilisation:** This function calculates and visualises the utilisation of the OTs based on the data on the performed surgeries. It combines the resource data across multiple simulation runs and calculates each day's utilisation. The function then creates boxplots per scheduled day for each OT and for the hospital in total.

**overtime\_visualization:** This function creates three plots per OT. One that visualises the frequency of overtime occurrences for each scheduling day using a boxplot, one that visualises the length of overtime occurrences for each schedule using a boxplot, and one that visualises the percentages of days in the scheduling cycle that experience a certain length of overtime using a histogram. It then saves the plot to the designated directory.

**Unavailability:** This function uses the information on the surgeries that are cancelled either due to ward unavailability or OT unavailability and counts for each ward or OT how many times they caused a surgery to be cancelled. Also, calculate the average, variance and standard deviation. It exports this information to an Excel file.

**Packages:**

* Pandas
* numpy
* matplotlib.pyplot
* seaborn

# Ward OT scheduling

This file contains several functions to set up the correct schedules. It includes the information of what the capacity should be for each of the wards and the OTs at each time step and returns it. It also repeats the number of schedules that are indicated.

**Ick\_value**: returns the capacity value for the indicated day and type of ICK. (used to set column in a data frame for ICK capacity)

**MCU\_value**: returns the capacity value for the indicated day, time and department of MCU. (used to set column in a data frame for MCU capacity, not used for KCN)

**KCN\_value**: returns the capacity value for the indicated day and time for the KCN. This has differences between the day that is separate from the others so it gets its own function. (used to set column in a data frame for KCN capacity)

**daycare\_value**: returns the capacity value for the indicated day and time of daycare. (used to set column in a data frame for daycare capacity)

**OT\_availablity**: returns the capacity for the OT for the indicated day and time. The capacity for each OT is one so it is to indicate when the OT is opened and closed. Setup across all the Ots

**Load\_json**: returns the cells of the data frame for schedules in the correct format, string

**Returns**: Data frame of ward capacity, data frame of OT capacity

**Packages:**

* Pandas
* json

# Input variable creation

This directory contains several Python files that contribute to the creation of the model data.

## Assigning group numbers

This file uses the clean data on the ward stays and OT time and an Excel file containing which surgeries belong to which grouping to combine the datasets to assign a group to each surgery and, with that, a surgery time and length of stay.

**Output:** Excel file with group numbers and categories assigned to each entry.

**Packages:**

* Pandas

## Distribuitons length of stay

This file creates plots of the length of stay per grouping and uses tests like KS, chi-squared and QQ-plots to assign a distribution to each grouping. Using the information data frame from the Assigning group numbers file as input.

**Output:** An Excel file with the results of the different tests and an Excel file including the distributions per group, shape and scale parameter and the average length of stay for each group.

**Packages:**

* Pandas
* Numpy
* Matplotlib.pyplot
* seaborn

## Distribuitons surgery duration

This file creates plots of the surgery durations per grouping and uses tests like KS, chi-squared and QQ-plots to assign a distribution to each group. Using the information data frame from the Assigning group numbers file as input.

**Output:** An Excel file with the results of the different tests and an Excel file including the distributions per group, shape and scale parameters and the average surgery duration for each group.

**Packages:**

* Pandas
* Numpy
* Matplotlib.pyplot
* seaborn

## functions distributions

The setup of the distributions uses several functions. These are set up in this file.

fit\_distributions: These fit the requested distribution based on the inputted data. It returns the parameters for the distribution.

Plot\_qq: Uses the data, chosen distribution and the fitted parameters to create a QQ plot. These QQ plots are returned.

Chi\_suared\_tests: Uses the data, chosen distribution and the fitted parameters to perform a chi-square test. These chi-statistics and the p-value are returned.

Ks\_test: Uses the data, chosen distribution and the fitted parameters to perform a KS-test. These KS statistics and the p-value are returned.

**Packages:**

* Numpy
* Statsmodels.api
* Scipty.stats

## probabilities

Takes the output data frames for both distributions of the length of stay and surgery duration and combines it with the probability of IC assignment for each grouping. It then returns a data frame with all parameters, averages and probabilities that can be used as input for the model. (run only after fitting de distributions for the length of stay and surgery duration)

**Packages:**

* pandas

## total dataset

Creates the plots for the total dataset for both surgery duration and length of stay. It also fits the different distributions, performs the KS-test chi-squared test and creates a QQ-plot.

# Input data

Contains the data that is used as input for the operations performed by the files in the directory input data, including:

* Clean\_data\_OR+Ward.xlsx: Contains instances of surgeries, including the duration of surgery, corresponding length of stay and assignment of test or training data. (This version is not uploaded to the GitHub as it contains patient data, it is replaced by Fake\_Medical\_Dataset\_with\_Namen, in order to illustrate what the dataset should look like)
* IC\_probabilities.xlsx: The probability of IC assignment calculated using the occurrences of the number of each surgery in each grouping that has either a 100% or 30% probability of being assigned an IC after surgery
* Surger\_Groups.xlsx: Contains the surgery names and to which group and department they belong
* (the files created by input variable creation that are not directly for the model use)

# Model data

This directory contains all the files used as input for the model, including eh Distributie\_parameters created in the input variable creation and the schedules.

# results

This directory is used to store all the output of the model. The file already comes with the base runs for all four schedules, the validation tests and the sensitivity analysis.