parsing_edgetimes_excel

Milla Juntunen

2022-06-29

```
library("readxl")
library("dplyr")
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
parse_file.function <- function(file_location, sheet_name) {

  # Reading the file and deleting empty rows
  df <- na.omit(read_excel(file_location, sheet_name))
  print(df)

  # Making empty list
  list_data <- list()

  # Counting rows
  rows = nrow(df)
  row <- 1

  # Reading the excel row by row
  for (row in 1:rows){

    # Reading data from a row
    row_data <- df[row,]

    # Parsing the Excel:

    # Defining the edge between nodes in flowchart and/or excel
    from <- as.character(select(row_data, from))
    to <- as.character(select(row_data, to))
    edge_name <- paste(from, to, sep = " - ")

    # Limits for the times between events
    lower <- as.numeric(select(row_data, start))
    upper <- as.numeric(select(row_data, end))
```

```
    # Probability for the edge
    probability <- as.numeric(select(row_data, probability))

    # Listing parsed information
    list_data[row] <- list(c(from, to, edge_name, lower, upper, probability))

  }
  return(list_data)
}
```

```
draw_vehicle.function <- function(from){

  # Vectors of which vehicles go to the scene and which are used to transport the patient   to the hosp
  vehicles_to <- c("Ambulance", "Ambulance and doctor", "Helicopter", "Fire truck")
  vehicles_from <- c("Ambulance", "Ambulance and doctor", "Helicopter")

  # Probabilities for each vehicle used
  probs_to <- c(0.4, 0.3, 0.25, 0.05)
  probs_from <- c(0.5, 0.3, 0.2)

  # Drawing a vehicle depending on a current node
  if(from == "Risk analysis"){
    vehicle <- as.character(sample(vehicles_to, 1, replace = FALSE, probs_to))
    cat(vehicle, " is the vehicle to rush to the scene!\n")
  } else if (from == "Packing"){
    vehicle <- as.character(sample(vehicles_from, 1, replace = FALSE, probs_from))
    cat(vehicle, " is the vehicle to rush to the hospital!\n")
  }
  return(vehicle)
}
```

```
check_vehicle.function <- function(vehicle, list_data, from){

  # Going through list_data to make a correct connection between nodes and to find a match between vehi
  # Should it be while loop instead?
  for (i in 1:length(list_data)){

    row <- list_data[i]
    data_from <- as.character(lapply(row,'[[',1))
    data_to <- as.character(lapply(row,'[[',2))

    # comparing data_to and vehicle if it is a match
    comp <- grepl(vehicle, data_to)

    # Must check that the correct vehicle is in the "to"
    if(data_from == from && comp){
      comp_check <- grepl("Ambulance and doctor", data_to, fixed = TRUE)

      if(vehicle == "Ambulance" && comp_check == FALSE){
        to <- data_to
        #cat(to, "is the place where Ambulance is going\n")
        #break
      } else if (vehicle != "Ambulance"){
      to <- data_to
```

```
      }
    }
  }
  # Connecting from and to to make a edge between nodes in the chart
  ret <- c(from, to)
  return(ret)
}
```

```
time_calculator.function <- function(from_to, total_time_count, infusion_time_count, list_data, infusion

  # Calculating the time between events (nodes)

  #stop_sign <- "Shock Room"
  need_vehicle_check <- TRUE
  from <- from_to[1]
  to <- from_to[2]

  # Going through list_data
  for (i in 1:length(list_data)){
    # Checking if it is time to stop and exit the loop
    #if(from == stop_sign) {break}

    # Checking if vehicle check is needed
    if(from == "Packing" && need_vehicle_check){

      # Must draw (new) vehicle for transportation
      vehicle <- draw_vehicle.function(from)
      from_to <- check_vehicle.function(vehicle, list_data, from)

      from <- from_to[1]
      to <- from_to[2]
      need_vehicle_check <- FALSE
    }

    # If chart B, must check that 'to' is correct after 'infusion starts' (matches vehicle)
    # How does this function know if we're using chart_a or chart_b
    # Could always pass a variable :'D
    #doesn't work at this point!
    if (chart_to_follow == "chart b" && from == "Infusion starts"){
      from_to <- check_vehicle.function(vehicle, list_data, from)
      from <- from_to[1]
      to <- from_to[2]
    }

    # Getting the "from" and "to" from the data list for comparison
    row <- list_data[i]
    data_from <- as.character(lapply(row,'[[',1))
    data_to <- as.character(lapply(row,'[[',2))

    # Comparing the datas. If those match, then getting the distribution limits.
    if(data_from == from && data_to == to) {
      lower <- as.numeric(lapply(row,'[[',4))
      upper <- as.numeric(lapply(row,'[[',5))
```

```
    # Updating the total_time_count
    random_time <- runif(1, lower, upper)
    total_time_count <- total_time_count + random_time

    # Checking if the infusion has started
    if(from == "Infusion starts"){
      infusion_starts <- TRUE
    }

    #If infusion has started, updating also infusion_time_count
    if(infusion_starts == TRUE) {
      infusion_time_count <- infusion_time_count + random_time
    }

    # Some printing to see if this works at all
    cat("\n")
    cat(as.character(lapply(row,'[[',3)),"\n")
    cat(lower, " is the lower limit for the time distribution\n")
    cat(upper, " is the upper limit for the time distribution\n")
    cat(random_time, " is the randomized time between limits\n")
    cat(total_time_count, " is the total time that has passed\n")
    cat(infusion_time_count, " is the time that has passed since the infusion started\n\n")

    # "to" is the next "from"
    from <- to

    # Getting the next node, this is not the best way to do it
    # Must find the other "to", done below. Checking if multiple next nodes exists and drawing one wi

    match_count <- 0
    possible_to <- list()
    probs <- list()

    for (j in 1:length(list_data)){
      row <- list_data[j]
      data_from <- as.character(lapply(row,'[[',1))

      if(data_from == from){

        # Match found, so increasing the match_count and saving the "to" in a list
        match_count <- match_count + 1

        to <- as.character(lapply(row,'[[',2))
        possible_to <- append(possible_to, to)

        prob <- as.numeric(lapply(row,'[[',6))
        probs <- append(probs, prob)

        # Drawing to from the list "possible_to"
        if(match_count > 1){
          to <- as.character(sample(possible_to, 1, replace = FALSE, probs))
        }
      }
```

```
      }
    }
  }
  ret <- c(from, to, total_time_count, infusion_time_count, infusion_starts)
}
```

```
### MAIN FILE STARTS HERE

infusion_starts <- FALSE
total_time_count <- 0
infusion_time_count <- 0
file_location <- "C:\\Projektit\\whole_blood_research\\excel\\EmergencyProcess_EdgeTimes.xlsx"
sheet_name_a <- "test_times_a"
sheet_name_b <- "test_times_b"
from <- "Risk analysis"

# Reading the file, listing information from the sheets
list_data_a <- parse_file.function(file_location, sheet_name_a)
```

```
## # A tibble: 21 x 5
##    from                          to                       start   end probability
##    <chr>                         <chr>                    <dbl> <dbl>       <dbl>
##  1 Risk analysis                 Ambulance leaving            1   1.5        0.4
##  2 Risk analysis                 Ambulance and doctor le~     1   1.5        0.3
##  3 Risk analysis                 Helicopter leaving           2   5          0.25
##  4 Risk analysis                 Fire truck leaving           1   1.5        0.05
##  5 Ambulance leaving             On the scene                20  40          1
##  6 Ambulance and doctor leaving  On the scene                20  40          1
##  7 Helicopter leaving            On the scene                10  20          1
##  8 Fire truck leaving            On the scene                20  40          1
##  9 On the scene                  Meeting the patient          0  10          1
## 10 Meeting the patient           Infusion starts              1  20          1
## # ... with 11 more rows
```

```
list_data_b <- parse_file.function(file_location, sheet_name_b)
```

```
## # A tibble: 26 x 5
##    from                          to                       start   end probability
##    <chr>                         <chr>                    <dbl> <dbl>       <dbl>
##  1 Risk analysis                 Ambulance leaving            1   1.5        0.4
##  2 Risk analysis                 Ambulance and doctor le~     1   1.5        0.3
##  3 Risk analysis                 Helicopter leaving           2   5          0.25
##  4 Risk analysis                 Fire truck leaving           1   1.5        0.05
##  5 Ambulance leaving             On the scene                20  40          1
##  6 Ambulance and doctor leaving  On the scene                20  40          1
##  7 Helicopter leaving            On the scene                10  20          1
##  8 Fire truck leaving            On the scene                20  40          1
##  9 On the scene                  Meeting the patient          0  10          1
## 10 Meeting the patient           Packing                      1  20          1
## # ... with 16 more rows
```

```
#cat("Printing listed data from chart_a (infusion starts on the scene)\n\n")
#print(list_data_a)

#cat("Printing listed data from chart_b (infusion starts during transport)\n\n")
```

```r
#print(list_data_b)

set.seed(2)

# Drawing whether the infusion starts on the scene (chart_a) or during transportation (chart_b)
charts <- c("chart a", "chart b")
chart_to_follow <- charts[sample(1:length(charts),1)]
if(chart_to_follow == "chart a"){
  list_data <- list_data_a
} else {list_data <- list_data_b}
cat(chart_to_follow, "is the chart to follow\n")
```

```
## chart a is the chart to follow
```

```r
# Drawing the vehicle to go to the scene
vehicle <- draw_vehicle.function(from)
```

```
## Helicopter  is the vehicle to rush to the scene!
```

```r
# Getting the first "time-between-events" edge from the function, also checking if the vehicle is okay
from_to <- check_vehicle.function(vehicle, list_data, from)

# Starting the time-calculator
current_stage <- time_calculator.function(from_to, total_time_count,infusion_time_count, list_data, infu
```

```
##
## Risk analysis - Helicopter leaving
## 2  is the lower limit for the time distribution
## 5  is the upper limit for the time distribution
## 3.719979  is the randomized time between limits
## 3.719979  is the total time that has passed
## 0  is the time that has passed since the infusion started
##
##
## Helicopter leaving - On the scene
## 10  is the lower limit for the time distribution
## 20  is the upper limit for the time distribution
## 11.68052  is the randomized time between limits
## 15.4005  is the total time that has passed
## 0  is the time that has passed since the infusion started
##
##
## On the scene - Meeting the patient
## 0  is the lower limit for the time distribution
## 10  is the upper limit for the time distribution
## 9.438393  is the randomized time between limits
## 24.83889  is the total time that has passed
## 0  is the time that has passed since the infusion started
##
##
## Meeting the patient - Infusion starts
## 1  is the lower limit for the time distribution
## 20  is the upper limit for the time distribution
## 18.92602  is the randomized time between limits
## 43.76492  is the total time that has passed
```

```
## 0  is the time that has passed since the infusion started
##
##
## Infusion starts - Packing
## 1  is the lower limit for the time distribution
## 5  is the upper limit for the time distribution
## 1.516636  is the randomized time between limits
## 45.28155  is the total time that has passed
## 1.516636  is the time that has passed since the infusion started
##
## Ambulance and doctor  is the vehicle to rush to the hospital!
##
## Packing - Ambulance and doctor leaving the scene
## 1  is the lower limit for the time distribution
## 2  is the upper limit for the time distribution
## 1.552674  is the randomized time between limits
## 46.83423  is the total time that has passed
## 3.06931  is the time that has passed since the infusion started
##
##
## Ambulance and doctor leaving the scene - At hospital
## 20  is the lower limit for the time distribution
## 40  is the upper limit for the time distribution
## 24.7779  is the randomized time between limits
## 71.61212  is the total time that has passed
## 27.84721  is the time that has passed since the infusion started
##
##
## At hospital - Shock Room
## 1  is the lower limit for the time distribution
## 10  is the upper limit for the time distribution
## 7.84462  is the randomized time between limits
## 79.45674  is the total time that has passed
## 35.69182  is the time that has passed since the infusion started
```

```
# Printing information from the calculators
total_time_count <- as.numeric(current_stage[3])
infusion_time_count <- as.numeric(current_stage[4])
cat("Patient transported to Shock Room!\n")
```

```
## Patient transported to Shock Room!
```

```
cat(total_time_count, "is the total time from the 'Risk Analysis'\n")
```

```
## 79.45674 is the total time from the 'Risk Analysis'
```

```
cat(infusion_time_count, "is the total time from the 'Infusion Starts'\n\n")
```

```
## 35.69182 is the total time from the 'Infusion Starts'
```