simid_workshop_tutorial.R

lwillem

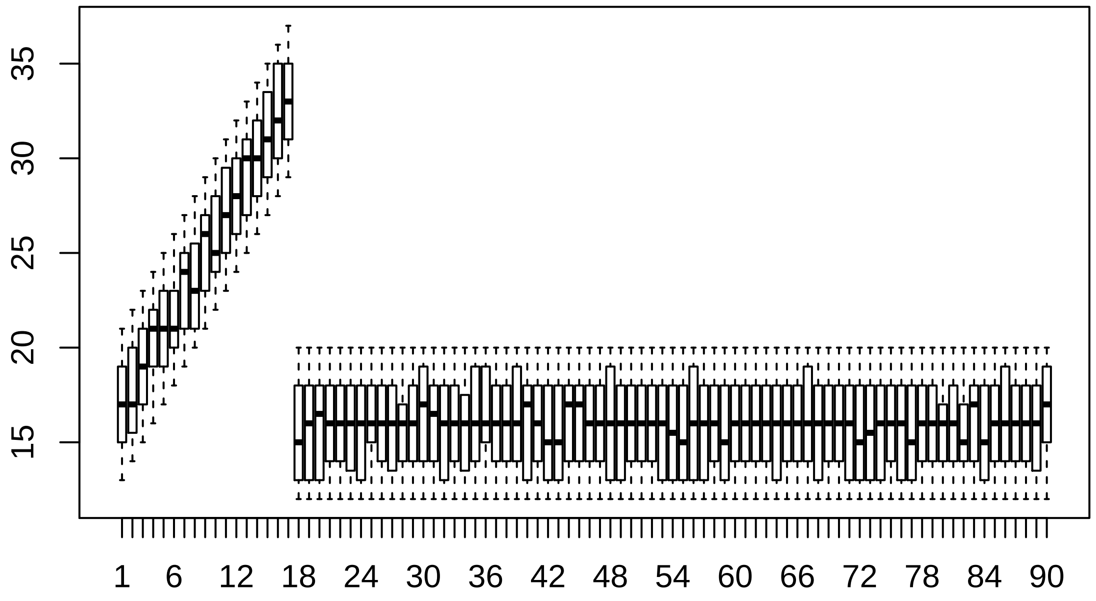
Fri May 24 10:34:14 2019

```r
############################################################################
# SIMID TUTORIAL: GENERAL ISSUES
#
# This program is free software: you can redistribute it and/or modify
# it under the terms of the GNU General Public License as published by
# the Free Software Foundation, either version 3 of the License, or
# (at your option) any later version.
#
# Copyright (C) 2019 lwillem, SIMID, UNIVERSITY OF ANTWERP, BELGIUM
############################################################################

# clear workspace
rm(list=ls())

# # set working directory (or open RStudio with this script)
# setwd("C:\User\path\to\the\rcode\folder") ## WINDOWS
# setwd("/Users/path/to/the/rcode/folder") ## MAC


# ###################
#  SETUP
# ###################
print('SETUP PARAMETERS: START')
```

```
## [1] "SETUP PARAMETERS: START"
```

```r
# set random number generator
rng_seed <- 20190524
set.seed(rng_seed)

# e.g. population details
pop_size      <- 1e4
max_age       <- 90
adult_age     <- 18
gender_male   <- 'M'
gender_female <- 'F'
gender_opt    <- c(gender_male, gender_female)
zipcode_opt   <- c(1000,2160,2640,2018,2110,3000,3500,3520,3590,9000)
community_opt <- 1:30

# e.g. social contact details
mean_num_cnt  <- 16          # population average
dev_num_cnt   <- -4:4        # deviance
```

```r
# e.g. create population data
pop_data <- data.frame(id        = 1:pop_size,
                       age       = sample(1:max_age,pop_size,replace = T),
                       gender    = sample(gender_opt,pop_size,replace = T),
                       zipcode   = sample(zipcode_opt,pop_size,replace = T),
                       community = sample(community_opt,pop_size,replace = T))

# e.g. sample number of contacts, based on population average
pop_data$num_contacts <- mean_num_cnt + sample(dev_num_cnt,pop_size,replace=T)

# e.g. children make more contacts, relative to their age
is_child                        <- pop_data$age<adult_age
pop_data$num_contacts[is_child] <- pop_data$num_contacts[is_child] +
                                     pop_data$age[is_child]

# explore "number of contacts"
boxplot(num_contacts ~ age, data = pop_data)
```



```r
print('SETUP PARAMETERS: COMPLETE')
```

```
## [1] "SETUP PARAMETERS: COMPLETE"
```

```r
# ################### #
#  SPEED              #
# ################### #
print('SPEED TESTS: START')
```

```
## [1] "SPEED TESTS: START"
```

```r
# for-loop
pop_data$is_male_adult <- FALSE
for(i_person in 1:pop_size){
  if(pop_data$age[i_person] >= adult_age & pop_data$gender[i_person] == gender_male){
    pop_data$is_male_adult[i_person] <- TRUE
  }
}

# vector operation
pop_data$is_male_adult <- pop_data$age >= adult_age & pop_data$gender == gender_male

# define help function
is_adult <- function(x,adult_age){
  return(x>=adult_age)
  }

# use 'lapply'
dummy <- unlist(lapply(pop_data$age,is_adult,adult_age))

# use vector operation
dummy <- pop_data$age >= adult_age
dummy <- is_adult(pop_data$age,adult_age)

print('SPEED TESTS: COMPLETE')
```

```
## [1] "SPEED TESTS: COMPLETE"
```

```r
# ###################
#  RANDOM NUMBERS
# ###################
print('RANDOM NUMBER SECTION: START')
```

```
## [1] "RANDOM NUMBER SECTION: START"
```

```r
# SITUATION 1: seed RNG once
set.seed(rng_seed)
age_once          <- sample(1:max_age,pop_size,replace = T)
zipcode_once      <- sample(zipcode_opt,pop_size,replace = T)

# SITUATION 2: seed RNG twice with same value
set.seed(rng_seed)
age_twice      <- sample(1:max_age,pop_size,replace = T)
set.seed(rng_seed)
zipcode_twice  <- sample(zipcode_opt,pop_size,replace = T)

# SITUATION 3: seed RNG multiple times with different seeds
set.seed(rng_seed)
age_multiple      <- sample(1:max_age,pop_size,replace = T)
set.seed(rng_seed+1)
zipcode_multiple  <- sample(zipcode_opt,pop_size,replace = T)

# PEARSON CORRELATION COEFICIENT??
abs(cor(age_once,zipcode_once))
```

```
## [1] 0.008595277
```

```r
abs(cor(age_twice,zipcode_twice))
```

```
## [1] 0.7697025
```

```r
abs(cor(age_multiple,zipcode_multiple))  # reproducibility?
```

```
## [1] 0.005513548
```

```r
print('RANDOM NUMBER SECTION: COMPLETE')
```

```
## [1] "RANDOM NUMBER SECTION: COMPLETE"
```

```r
# ###################
#  PARAMETER SWEEP
# ###################
print('PARAMETER SWEEP: START')
```

```
## [1] "PARAMETER SWEEP: START"
```

```r
# get all combinations
exp_design_grid <- expand.grid(age       = 1:max_age,
                               gender    = gender_opt,
                               zipcode   = zipcode_opt,
                               community = community_opt)
dim(exp_design_grid)
```

```
## [1] 54000     4
```

```r
# load package: latin hypercube sampling
library(lhs)

num_param <- 4
num_exp   <- 10000

# get Latin Hypercube design
exp_design_maximin <- maximinLHS(num_param,num_exp)

# get extended Latin Hypercube design
exp_design_maximin_extended <- maximinLHS(num_param*40,num_exp)
exp_design_random_extended  <- randomLHS(num_param*40,num_exp)

# TODO: tranfer LHD into parameters...
# send an email to lander.willem@uantwerp.be :-)

print('PARAMETER SWEEP: COMPLETE')
```

```
## [1] "PARAMETER SWEEP: COMPLETE"
```

```r
# ###################
#  COPY/PASTE...
# ###################
print('COPY/PASTE: START')
```

```
## [1] "COPY/PASTE: START"
```

```r
# e.g. column names
col_names <- c('member1','member2','member3','member4','member5','member6','member8',
               'member9','member10','member11','member12','member13')
length(col_names)
```

```
## [1] 12
```

```r
# e.g. column names ==> using 'paste'
col_names <- paste0('member',1:13)
length(col_names)
```

```
## [1] 13
```

```r
print('COPY/PASTE: COMPLETE')
```

```
## [1] "COPY/PASTE: COMPLETE"
```

```r
# ###################
# COLUMN NAMES...
# ###################
print('COLUMN NAMES: START')
```

```
## [1] "COLUMN NAMES: START"
```

```r
names(pop_data)
```

```
## [1] "id"            "age"           "gender"        "zipcode"
## [5] "community"     "num_contacts"  "is_male_adult"
```

```r
# e.g. get mean age
mean(pop_data[,2])
```

```
## [1] 45.4494
```

```r
# ... somewhere in the code... or the original data changes...
pop_data <- cbind(1,pop_data)

# get mean age ??
mean(pop_data[,2])
```

```
## [1] 5000.5
```

```r
mean(pop_data$age)
```

```
## [1] 45.4494
```

```r
print('COLUMN NAMES: COMPLETE')
```

```
## [1] "COLUMN NAMES: COMPLETE"
```