

# Modelling Infectious Diseases using R

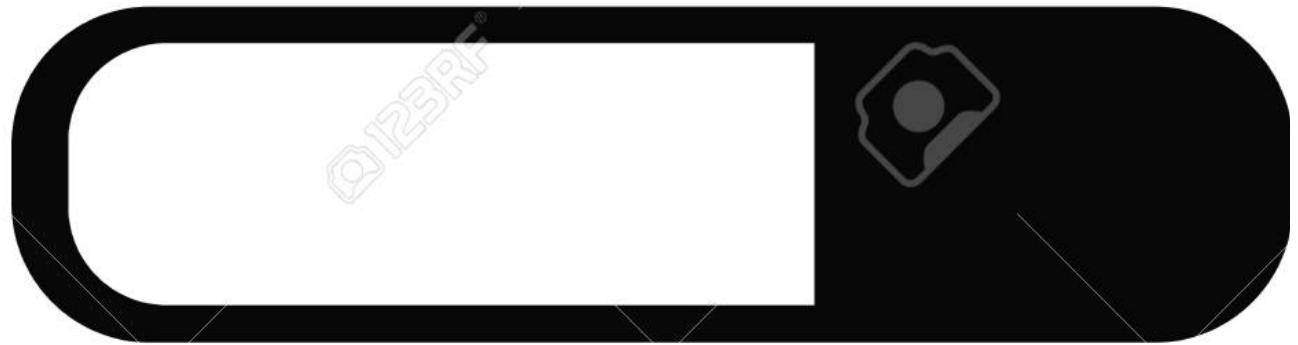
Lander Willem  
SIMID workshop 2019



**Health Economics & Modelling Infectious Diseases**  
**Vaccine & Infectious Disease Institute**  
University of Antwerp



# USER EXPERIENCE



loading . . .



# **[https://github.com/lwillem/simid\\_rtools](https://github.com/lwillem/simid_rtools)**

```
#####
# 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
# OR: use the directory of the current file if you use "source"
```

```

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

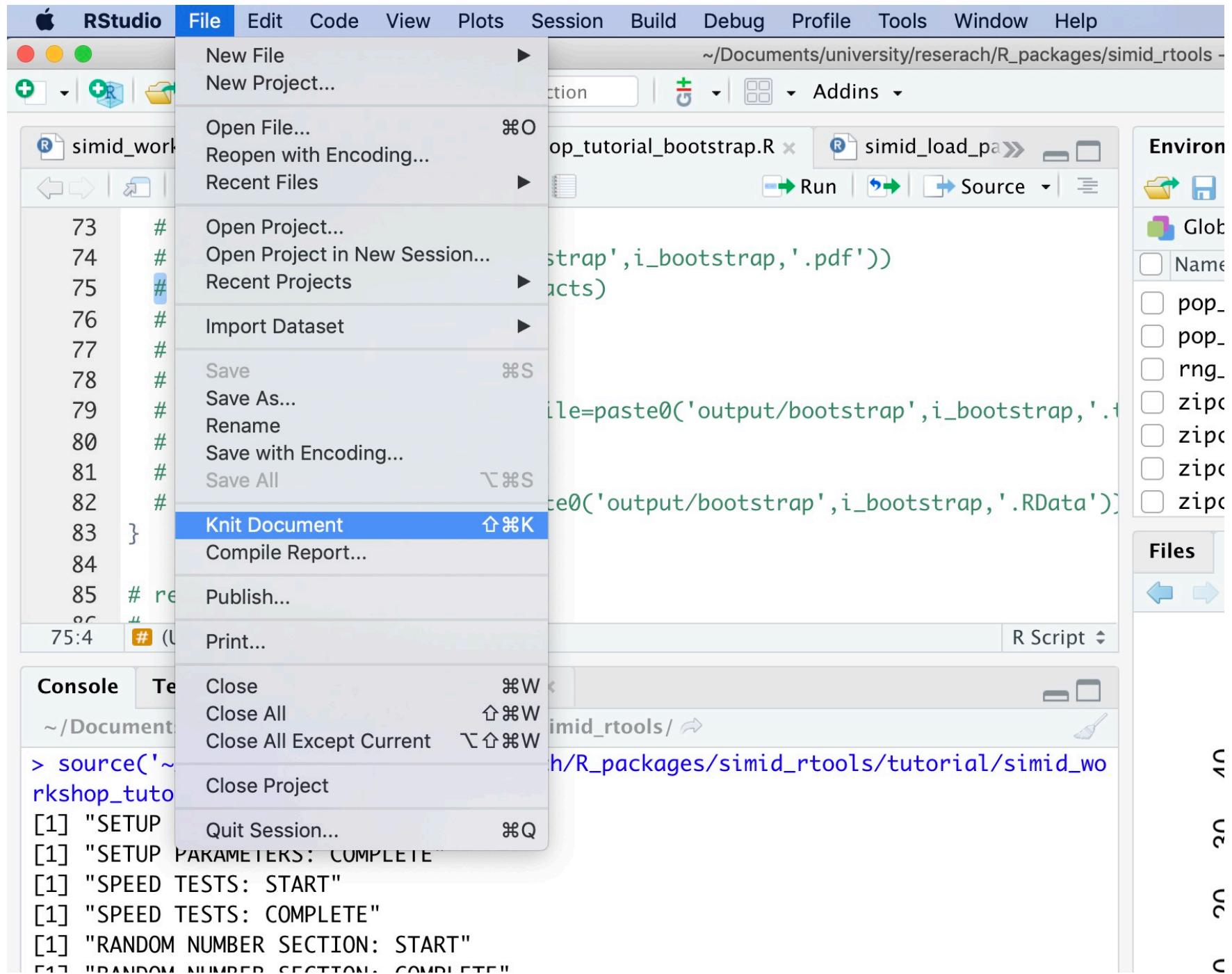
## [1] "SETUP PARAMETERS: START"

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

# e.g. population details
pop_size      <- 1e4
mean_hh_size   <- 4
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. population data
pop_data <- data.frame(id           = 1:pop_size,
                        age            = sample(1:max_age,pop_size,replace = T),
                        gender         = sample(c(gender_male,gender_female),pop_size,replace = T),
                        zipcode        = sample(zipcode_opt,pop_size,replace = T),

```



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

## [1] "SPEED TESTS: START"

# 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
```

**pop\_size = 10 000**  
**Difference?**  
**Factor?**

RStudio File Edit Code View Plots Session Build Debug Profile Tools Window Help

~/Documents

simid\_workshop\_tutorial.R x simid\_workshop\_tutorial\_bootstrap.R

Source on Save

52 # ##### #  
53 # SPEED #  
54 # ##### #  
55 print('SPEED TESTS: START')  
56  
57 # for-loop  
58 pop\_data\$is\_male\_adult <- FALSE  
59 for(i\_person in 1:pop\_size){  
60 if(pop\_data\$age[i\_person] >= adult\_age & pop\_data\$gender[i\_person] == gender\_male){  
61 pop\_data\$is\_male\_adult[i\_person] <- TRUE  
62 }  
63 }  
64  
65 # vector operation  
66 pop\_data\$is\_male\_adult <- pop\_data\$age >= adult\_age & pop\_data\$gender == gender\_male  
67  
68 # define help function  
69 is\_adult <- function(x,adult\_age){  
70 return(x>=adult\_age)  
71 }  
72  
73 # use 'lapply'  
74 dummy <- unlist(lapply(pop\_data\$age,is\_adult,adult\_age))  
75  
76 # use vector operation  
77 dummy <- pop\_data\$age >= adult\_age  
78 dummy <- is\_adult(pop\_data\$age,adult\_age)|

Start Profiling  
Stop Profiling  
Profile Selected Line(s) ⌘⌃⌘P  
Open Profile...  
Profiling Help

d\_rtools - master - RStudio

misc\_package\_contact

7

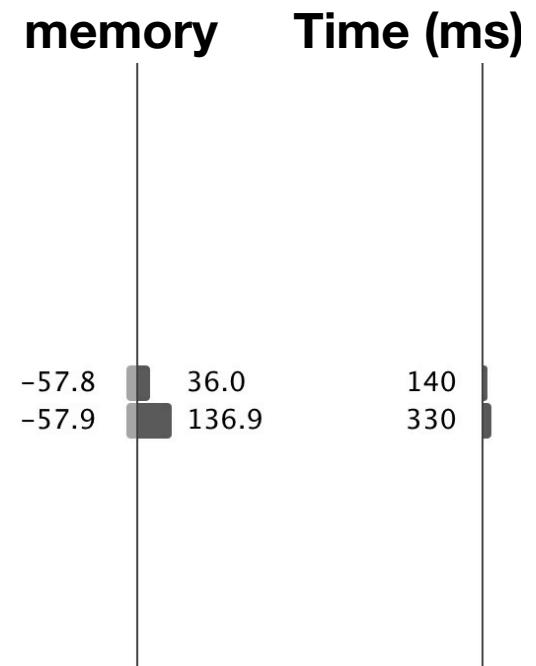
```

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

# 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

```



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

```
# 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)
```

**pop\_size = 10 000**  
**Difference?**  
**Factor?**

```

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

```

```

# 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)

```

**memory**      **Time (ms)**

2.6      10

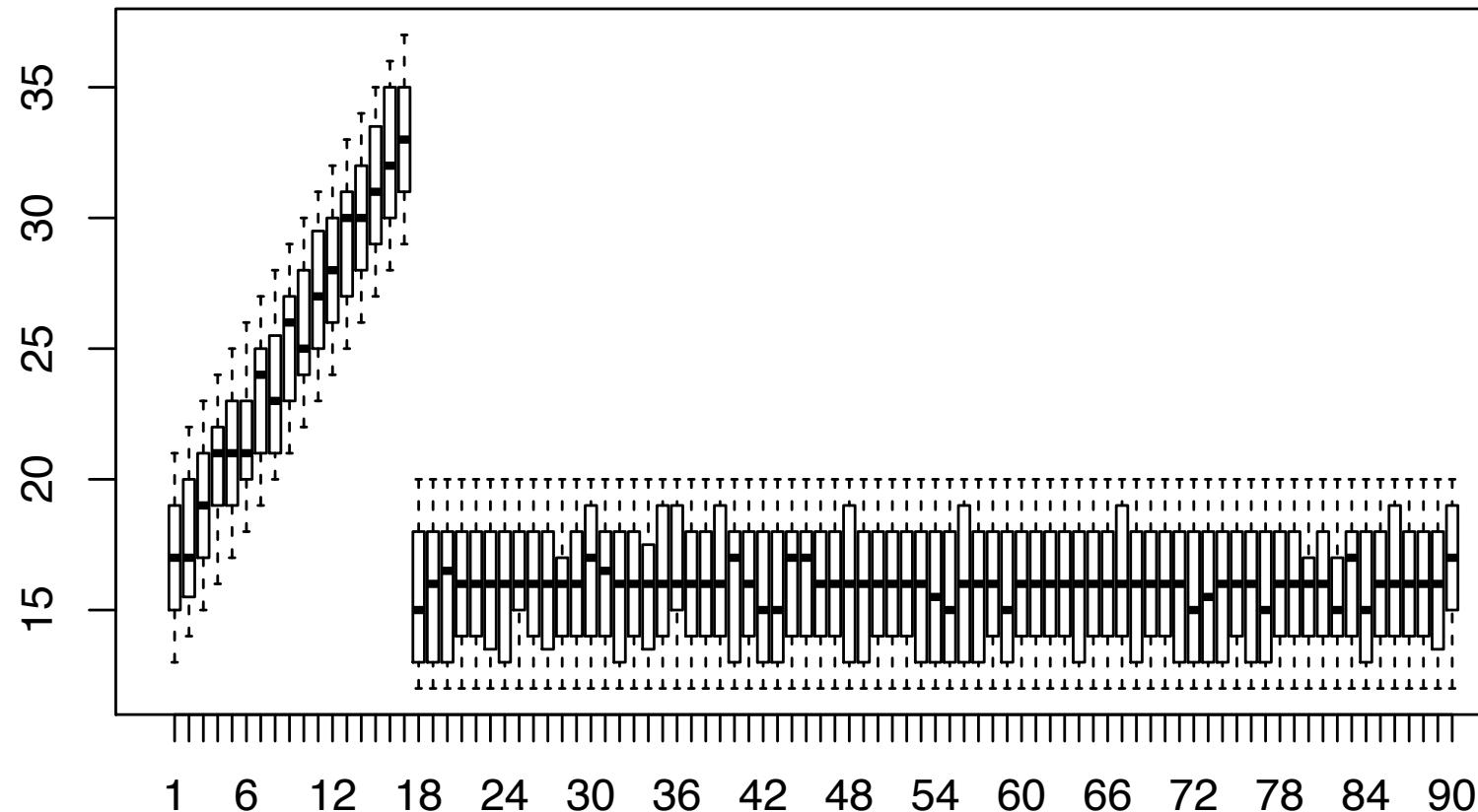
```

# e.g. dependent variable: sample number of contacts
pop_data$num_contacts <- 16 + sample(-4:4,pop_size,replace=T)

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

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

```



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

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

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

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

```
# 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

abs(cor(age_twice,zipcode_twice))

## [1] 0.7697025

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

## [1] 0.005513548
```

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

## [1] "PARAMETER SWEEP: START"

# 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

# 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 14)
```

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

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



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



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

```
# e.g. column names
```

```
col_names <- c('member1', 'member2', 'member3', 'member4', 'member5', 'member6', 'member8',  
             'member9', 'member10', 'member11', 'member12', 'member13')
```

```
length(col_names)
```

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

```
# e.g. column names ==> using 'paste'
```

```
col_names <- paste0('member', 1:13)
```

```
length(col_names)
```

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

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

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

```
# #####  
# COLUMN NAMES...  
# #####  
print('COLUMN NAMES: START')  
  
## [1] "COLUMN NAMES: START"  
  
names(pop_data)  
  
## [1] "id"           "age"          "gender"        "zipcode"  
## [5] "community"    "num_contacts"  "is_male_adult"  
  
# e.g. get mean age  
mean(pop_data[,2])  
  
## [1] 45.4494  
  
# ... 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  
  
mean(pop_data$age)  
  
## [1] 45.4494
```

~/Documents/university/reserach/R\_packages/simid\_rtools - master - RStudio

simid\_rtools

simid\_workshopTutorial.R simid\_workshopTutorial\_bootstrap.R simid\_load\_pa... Addins

Source on Save Run Source

```

73 # # plot results
74 # pdf(file=paste0('output/bootstrap',i_bootstrap,'.pdf'))
75 # plot(out_data_frame$num_contacts)
76 # dev.off()
77 #
78 # # save results to txt file
79 # write.table(out_data_frame,file=paste0('output/bootstrap',i_bootstrap,'.txt'))
80 #
81 # # save results to RData file
82 # save(out_data_frame,file=paste0('output/bootstrap',i_bootstrap,'.RData'))
83 }
84
85 # retrieve data etc
86

```

Environment History Connections Build Git

Import Dataset Global Environment

Name	Type	Length	Size	Value
pop_data	data.frame	0	0 B	10000 obs. of 8 varia...
pop_size	numeric	1	56 B	10000
rng_seed	numeric	1	56 B	20190524
zipcode_mult...	numeric	10000	78.2 KB	num [1:10000] 2110 3520...
zipcode_once	numeric	10000	78.2 KB	num [1:10000] 2018 2110...
zipcode_opt	numeric	10	176 B	num [1:10] 1000 2160 26...
zipcode_twice	numeric	10000	78.2 KB	num [1:10000] 3590 3000...

Files Plots Packages Help Viewer

Zoom Export Publish

Console Terminal R Markdown Jobs

~/Documents/university/reserach/R\_packages/simid\_rtools/

```

> source('~/Documents/university/reserach/R_packages/simid_rtools/tutorial/simid_workshopTutorial.R')
[1] "SETUP PARAMETERS: START"
[1] "SETUP PARAMETERS: COMPLETE"
[1] "SPEED TESTS: START"
[1] "SPEED TESTS: COMPLETE"
[1] "RANDOM NUMBER SECTION: START"
[1] "RANDOM NUMBER SECTION: COMPLETE"
[1] "PARAMETER SWEEP: START"
[1] "PARAMETER SWEEP: COMPLETE"
[1] "COPY/PASTE: START"
[1] "COPY/PASTE: COMPLETE"
[1] "COLUMN NAMES: START"
[1] "COLUMN NAMES: COMPLETE"
>

```

Console

Terminal ×

R Markdown ×

Jobs ×

~/Documents/university/reserach/R\_packages/simid\_rtools/ ↗

```
> source('~/Documents/university/reserach/R_packages/simid_rtools/tutorial/simid_workshop_tutorial.R')
[1] "SETUP PARAMETERS: START"
[1] "SETUP PARAMETERS: COMPLETE"
[1] "SPEED TESTS: START"
[1] "SPEED TESTS: COMPLETE"
[1] "RANDOM NUMBER SECTION: START"
[1] "RANDOM NUMBER SECTION: COMPLETE"
[1] "PARAMETER SWEEP: START"
[1] "PARAMETER SWEEP: COMPLETE"
[1] "COPY/PASTE: START"
[1] "COPY/PASTE: COMPLETE"
[1] "COLUMN NAMES: START"
[1] "COLUMN NAMES: COMPLETE"
>
```

```

# #####
# BOOTSTRAP: FOR-LOOP
# #####
print('BOOTSTRAPS: START')

## [1] "BOOTSTRAPS: START"
num_bootstraps <- 500

# for-loop

for(i_bootstrap in 1:num_bootstraps){

  # bootstrap population
  pop_bootstrap      <- pop_data[sample(pop_size,replace=T),]

  # get mean age by community and gender
  summary_num_contacts <- aggregate(num_contacts ~ age + gender, data = pop_bootstrap, mean)

  # aggregate results: data.frame
  out_data_frame      <- data.frame(i_bootstrap,
                                       summary_num_contacts)

  # # plot results
  # pdf(file=paste0('output/bootstrap',i_bootstrap,'.pdf'))
  # plot(out_data_frame$num_contacts)
  # dev.off()
  #

  # # save results to txt file
  # write.table(out_data_frame,file=paste0('output/bootstrap',i_bootstrap,'.txt'),row.names=F,sep=''
  #
  # # save results to RData file
  # save(out_data_frame,file=paste0('output/bootstrap',i_bootstrap,'.RData'))}

# retrieve data etc
# ....

```

```

# #####
# BOOTSTRAP: FOR-LOOP
# #####
print('BOOTSTRAPS: START')

```

```

## [1] "BOOTSTRAPS: START"
num_bootstraps <- 500

# for-loop 1
i_bootstrap <- 1
for(i_bootstrap in 1:num_bootstraps){


```

```

# bootstrap population
pop_bootstrap      <- pop_data[sample(pop_size,replace=T),]

# get mean age by community and gender
summary_num_contacts <- aggregate(num_contacts ~ age + gender, data = pop_bootstrap, mean)

# aggregate results: data.frame
out_data_frame     <- data.frame(i_bootstrap,
                                    summary_num_contacts)

```

```

# # plot results
# pdf(file=paste0('output/bootstrap',i_bootstrap,'.pdf'))
# plot(out_data_frame$num_contacts)
# dev.off()
#
# # save results to txt file
# write.table(out_data_frame,file=paste0('output/bootstrap',i_bootstrap,'.txt'),row.names=F,
# #
# # save results to RData file
# save(out_data_frame,file=paste0('output/bootstrap',i_bootstrap,'.RData'))}


```

```

# retrieve data etc
# ....

```

**Time (ms)**

2700

5310

100

450

810

540

600

320

**\*if not commented**

```

# ######
# SEQUENTIAL FOREACH
# #####
print('SEQUENTIAL FOREACH: START')

## [1] "SEQUENTIAL FOREACH: START"

library('foreach')

print(system.time( # start manual profiling
# for-loop 2: return values
bootstrap_out <- foreach(i_bootstrap = 1:num_bootstraps,
                           .combine      = 'rbind') %do%{

  # bootstrap population
  pop_bootstrap           <- pop_data[sample(pop_size,replace=T),]

  # get mean age by community and gender
  summary_num_contacts <- aggregate(num_contacts ~ age + gender, data = pop_bootstrap, mean)

  # aggregate results: data.frame
  out_data_frame          <- data.frame(i_bootstrap,
                                           summary_num_contacts)

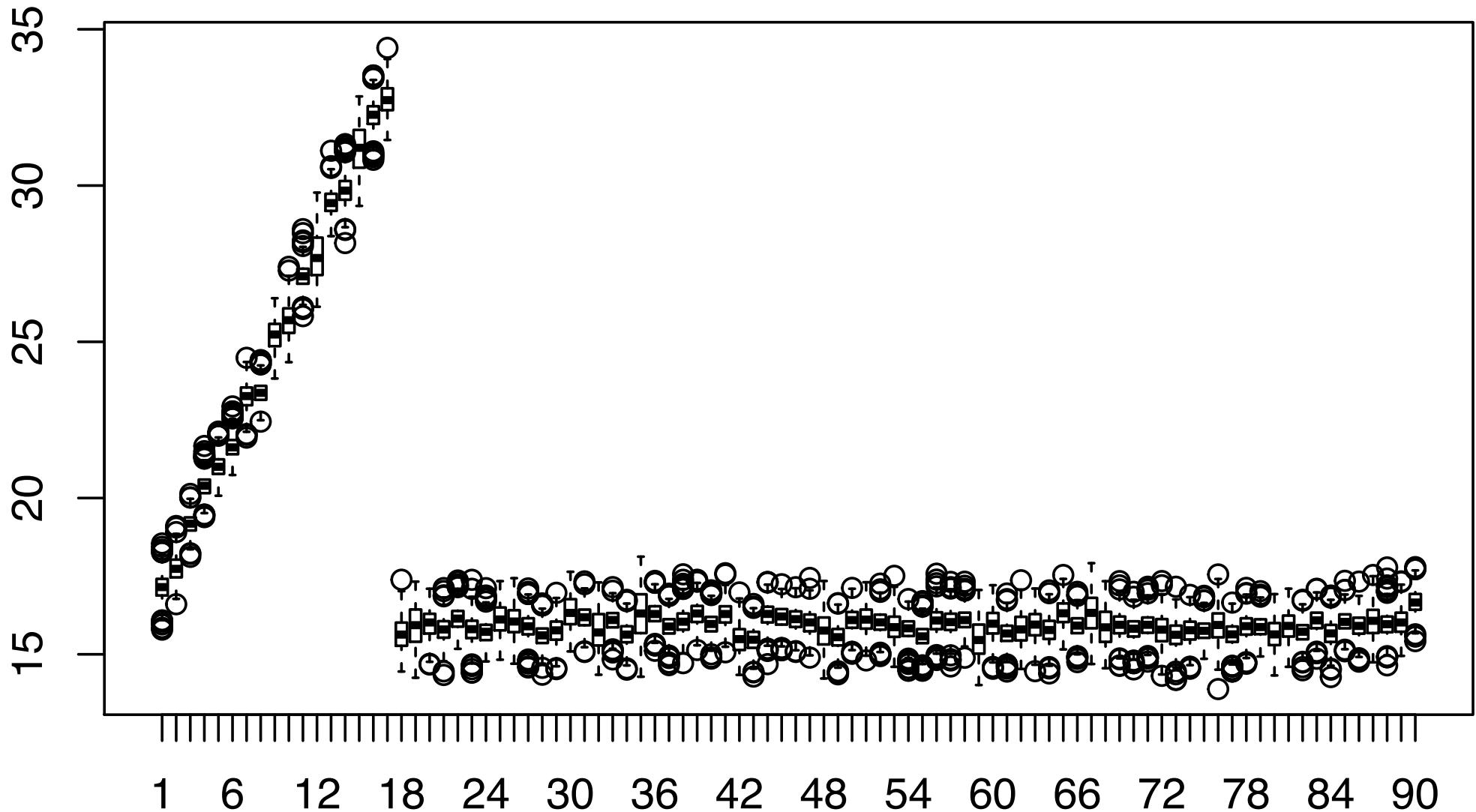
  # return results
  return(out_data_frame)
}
)) # end system.time()

##    user  system elapsed
##   6.100   1.155   7.269

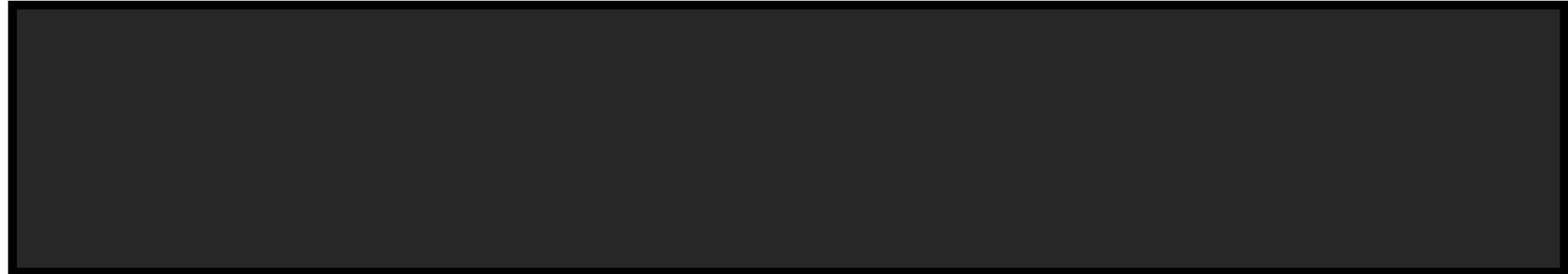
```

```
# plot bootstrap results
```

```
boxplot(num_contacts ~ age, data=bootstrap_out)
```



```
# #####  
# PARALLEL FOREACH  
# #####  
print('PARALLEL FOREACH: START')  
  
## [1] "PARALLEL FOREACH: START"  
# uncomment the following lines to install the simid.rtools package from github  
#install.packages('devtools')  
library(devtools)  
  
# install package "simid_rtools" from https://github.com/lwillem/simid_rtools  
devtools::install_github("lwillem/simid_rtools",force=F,quiet=T)  
  
# start parallel working nodes  
# note: make sure you loaded all required user defined functions at this point  
simid.rtools::smd_start_cluster()
```



```
simid.rtools::smd_stop_cluster()  
  
print('PARALLEL FOREACH: COMPLETE')  
  
## [1] "PARALLEL FOREACH: COMPLETE"
```

```

print(system.time( # manual profiling
# run loop in parallel
bootstrap_out <- foreach(i_bootstrap = 1:num_bootstraps,
                           .combine      = 'rbind' %dopar%{

    # bootstrap population
    pop_bootstrap           <- pop_data[sample(pop_size,replace=T),]

    # get mean age by community and gender
    summary_num_contacts <- aggregate(num_contacts ~ age + gender, data = pop_bo

    # aggregate results: data.frame
    out_data_frame          <- data.frame(i_bootstrap,
                                             summary_num_contacts)

    # return results
    return(out_data_frame)
}
)) # end system.time()

##     user   system elapsed
## 0.400   0.151   1.967

```

**Sequential:**

```

##     user   system elapsed
## 6.100   1.155   7.269

```

https://github.com/lwillem/simid\_rtools/commits/master

Search or jump to... Pull requests Issues Marketplace Explore

lwillem / simid\_rtools Unwatch 1 Star 0 Fork 0

Code Issues 0 Pull requests 0 Projects 0 Wiki Security Insights Settings

Branch: master

Commits on Apr 25, 2019

- smd\_file\_path update: check if a file extension is present before che... a4870cf
- Added function to get (and create) a folder path. a21108c
- Added package dependency for doParallel in the DESCRIPTION file + cos... 448b745
- Changed the 'smd\_progress' function so the master node is also allow... d799fb1

Commits on Mar 8, 2019

- Small changes to the foreach tutorial. 218c376
- Included a tutorial script and pdf. 01b441f
- Fixed issue with global par\_nodes\_info parameter. Not perfect yet... 7ecabc3
- Updated parallel admin functions and added function to load (and inst... 77e5878

*print("Thank you")*