## par\_foreach\_tutorial.R

## lwillem

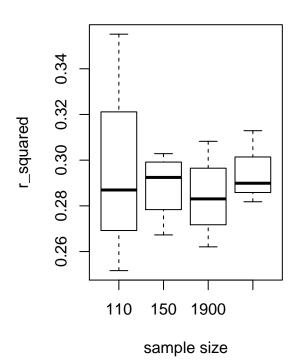
Fri Mar 8 14:38:45 2019

```
# SIMID TUTORIAL: PARALLEL FOREACH
# Default functions:
  - expand.qrid
                     to get all combinations of given parameters
  - foreach
                      to loop over a set (similar to 'for')
# New functions form the simid.rtools package
# - smd_load_packages to load (and install) packages
# - smd_start_cluster to start mutiple local processing nodes
  - smd_stop_cluster to end local work nodes
  - smd_progress to print the progress during the parallel loop
# 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())
# install the simid.rtools packge from github
devtools::install_github("lwillem/simid_rtools",force=F,quiet=T)
# load the package
library('simid.rtools')
# list all required packages for this project
all packages <- c('scales')</pre>
# load all packages (and install them if required)
smd_load_packages(all_packages)
# create a user defined (dummy) function
smd_sum <- function(x){</pre>
 return(sum(x))
# set up an experimental design
exp_design <- expand.grid( num_iter = seq(4),</pre>
                                                          # number of iterations
                        sample_size = c(110,150,1900,2500)) # sample_size
dim(exp_design)
```

## [1] 16 2

```
# start parallel working nodes
# note: make sure you loaded all required user defined functions at this point
par nodes info <- smd start cluster()</pre>
# print message to user, and store the current time (for the progress report)
smd_print('GRIDSEARCH...'); time_stamp <- Sys.time()</pre>
# run all experiments from the design
exp_results <- foreach(i_exp = 1:nrow(exp_design), # all experiments</pre>
                       .combine ='rbind') %dopar% {  # combine the output by row
   # print progress (only the first work-node)
   # note: the parallel environment requires to specify that we need the
   # 'smd_progress' function from the 'simid.rtools' package
   simid.rtools::smd_progress(i_exp,nrow(exp_design),time_stamp,par_nodes_info)
   # some dummy operations...
   x sample
             <- sample(seq(1,1000),exp_design$sample_size[i_exp],replace = T)</pre>
                 <- sample(seq(50,75),exp_design$sample_size[i_exp],replace = T)</pre>
   y_sample
   random_sample <- sample(seq(50:75),exp_design$sample_size[i_exp],replace = T)</pre>
   z_sample <- x_sample + y_sample + y_sample*random_sample</pre>
   # some fitting
   lm_model <- lm(z_sample ~ x_sample + y_sample)</pre>
   lm_summary <- summary(lm_model)</pre>
   # user defined function
   user_sum <- smd_sum(exp_design[i_exp,])</pre>
   # return loop index, parameters and results
   data.frame(id_iter
                        = i_{exp}
                                                     # run id
                                                     # input param
              exp_design[i_exp,],
              r_squared = lm_summary$r.squared, # output...
              user_sum = user_sum)
                                                     # output...
}
# terminate parallel cluster
smd stop cluster()
# explore output
par(mfrow=1:2)
boxplot(r_squared ~ sample_size,data=exp_results,
        xlab='sample size',ylab='r_squared',
        main='dummy output')
boxplot(user_sum ~ num_iter, data=exp_results,
        xlab='iteration',main='dummy output')
```

## dummy output



## dummy output

