```
##############################################################################################
# This program is provided for illustration purposes only and is not to be regarded as an
# official StandRe document.
#
# The program takes as input the IE1 model (mean frequency, probabilities for the dependent
# Bernoulli random variables, GenPareto severities per IE1 model segment, copula between severities)
# and generates from this the exceedance frequency curve of the total loss (severity summed up over
# all IE1 model segments).
#
# The exceedance frequency curve (efc) of the IE1 model is then compared to the exceedance frequency
# curve of the IE1 scenarios taken from the StandRe template (sheet
# "RE_IE1_calibr_to_scen_results_2" columns "severity" and "excess frequency"). The comparison is
# done graphically (incl. in log-log-#plot) as well as in a table at the scenario severities
# (in the R console).
#
# INPUT PARAMETERS for the IE1 model
#   n.simul   <- nr of simulations
#   nr.ms     <- nr of model segments
#   fc.theta  <- parameter of the flipped clayton copula for the model segment severities
#   prob.scen <- joint probabilities of the Bernoulli random variables
#
#   lambda   <- parameter for the mean frequency of the IE1 model
#   thres.ms,  alphai.ms, alphat.ms <- parameters of the IE1 model segment severities (Gen Pareto)
#
#  INPUT PARAMETERS for the IE1 scenarios
#   scen.sev <- vector of IE1 scenario severities
#   scen.efc <- vector of IE1 scenario exceedance frequencies
#
#  OUTPUT PARAMETERS
#
#  1) bernuolli.sim  <- realizations B_{i, l}  (sequence of 0 & 1's). Here, 1 at entry "l"
#                       corresponds to an event loss for model segment "l" being triggered by the
#                       simulated event "i"
#
#  2) loss.ms        <- realizations from the joint df of the aggregated event losses per
#                       IE1 mod seg
#
#  3) loss.total     <- realizations from the distribution of the total aggregated event loss
#                       (summed over IE1 model segm)
#
#  4) efc.IE1model   <- function that computes the efc of the IE1 model by using the estimated
#                       distribution function of the total event loss severities
#
#  5) Plot of the exceedance frequency curve of the total event loss severities and of the scenario
#     severities
#
#  6) table of exceedance frequencies at the scenario severities ( printed in the R console)
#
##--------------------------------------------------------------------------------------------

rm(list = ls())

library(fExtremes)  # allows to sample from generalized Pareto
library(copula)     # allows to sample from a given copula
require(knitr)      # allows printing the table

#------------------ Helper Functions ----------------------------------------------------------

# The function below generates the k-th binary number of length n,
# in increasing order where k = 0,..., 2^n-1

int.convert.Bin <- function (k, n)
{
  return(as.numeric((sapply(strsplit(paste(rev(intToBits(k))), "")[(32-n+1):32],`[[`, 2))))
}

# The function below generates a matrix with all potential binary combinations of length m in
# binary order (i.e. starting with (0,...,0,0,1); (0,...,0,1,0); (0,...,0,1,1); (0,...,1,0,0) etc.)

bin.matrix <- function (m)
{
  return(t(sapply(1:(2^{m}-1), int.convert.Bin, n = m)))
}

# The function below constructs a sample from a frequency-severity model, given samples
# of severities X and frequencies N

compound_poisson_construct <- function(X, N)
{
  n <- length(N)
  I <- as.factor(rep(1:n, N))
  S <- tapply(X, I, sum)
  V <- as.numeric(S[as.character(1:n)])
  V[is.na(V)] = 0
  return(V)
```

```
}

# The function below draws a discrete sample of size "nr.sim" from {1:length(prob.scen)}
# with probabilities: "prob.scen"

bernuolli.sample <- function(prob, nr.sim)
{
  if (sum(prob) == 1)
  {
    cdf_interval <- 1:length(prob)
    return(sample(unique(cdf_interval), nr.sim, replace = TRUE, prob = prob))
  } else
  {
    print("ERROR: probabilites do not sum up to 1")
  }
}

#------------------Parameters-----------------------------------------------------------------------

n.simul   <- 1e5   # nr of simulations
fc.theta  <- 0.5   # parameter theta of the (flipped) clayton copula, this parameter is fixed

# Company-specific parameters

nr.ms     <- 5     # nr of model segments
lambda    <- 1.8   # mean frequency for the IE1 model

prob.scen <- rep(1/(2^nr.ms-1), 2^nr.ms-1) # company-specific joint Bernoulli probabilities. It
                                           # corresponds to the values in column F (binary order)
                                           # of the sheet "RE_IE1_model_parameters" of the StandRe
                                           # template

# Parameters for the severities of model segments

thres.ms  <- c(1e6, 1.5e6, 0.4e6, 2e6, 2.5e6) # thresholds
alphai.ms <- c(1.5, 0.15, 1.3, 0.15, 0.4)   # alpha initial
alphat.ms <- c(1.5, 1.12, 2.1, 1.2, 3.1)           # alpha tail

#------------------Simulation of the frequency distribution ----------------------------------------

N         <- rpois(n.simul, lambda) # drawing a sample of size n.simul for the nr of event losses
matrix.bc <- bin.matrix(nr.ms)      # matrix containing the possible binary combinations
                                    # of length nr.ms, in binary order

# Sampling the Bernoulli realizations for each IE1 model segment given that an event occurred

hidden.sample <- bernuolli.sample(prob.scen, sum(N))
bernuolli.sim <- matrix(0.0, nrow = sum(N), ncol = nr.ms)
bernuolli.sim <- matrix.bc[hidden.sample, ] # contains all the simulations B_{i, l},
                                            # where l = 1,..., nr.ms and i = 1,..., N

# Side calculation: Implied expected frequency of severities to IE1 model segment "l" exceeding the
# modeling threshold "t_l"

ms.lambda <- rep(0, nr.ms)

for (i in 1:nr.ms)
{

  ms.lambda[i]  <-  lambda*sum(bernuolli.sim[, i] > 0)/sum(N)

}

#------------------Simulation of the model segment losses ------------------------------------------

# Initialization

vec.sev  <- matrix (0.0, ncol = nr.ms, nrow = sum(N))  # helper vector, contains all the simulations
                                                       #  X_{l,i}, where l = 1,...nr.ms and i = 1:N
loss.ms  <- matrix (0.0, ncol = nr.ms, nrow = n.simul) # losses per model segment level:
                                                       #  Sum_{i=1}^N B_{i,l}*X_{l,i}

# Dependence between the loss severities

copula    <- claytonCopula(fc.theta, dim = nr.ms)
unif.fc <- 1 - rCopula(sum(N), copula) #flipping the simulations obtained from clayton copula

# Generation of the losses (severities and aggregate) per IE1 model segment

for (i in 1:nr.ms)
{
  vec.sev[, i] <- rgpd(sum(N), xi = 1/alphat.ms[i], mu = thres.ms[i],
                     beta = thres.ms[i]/alphai.ms[i])
  vec.sev[, i] <- rev(sort(vec.sev[, i])[rank( unif.fc[, i] )]) #reordering the severities
  loss.ms[, i] <- compound_poisson_construct(vec.sev[, i]*bernuolli.sim[, i], N)
```

```r
}

#------------------Simulation of the total losses (severities and aggregate) -----------------------

sev.total  <- rowSums(vec.sev*bernuolli.sim)
loss.total <- rep(0, n.simul)
loss.total <- compound_poisson_construct(sev.total, N)    #rowSums(loss.ms)

#----------------------Estimated exceedance frequency curve for the IE1 model-----------------------

efc.IE1model <- function (severity, lambda)
{
  return(lambda*(1 - ecdf(sev.total)(severity)))
}

#----------------------Test case-----------------------------------------------------------

# Input parameters for the IE1 scenarios

# Vector of total severity per IE1 scenario

scen.sev <- c(14950000, 9388000, 18296000, 13076000, 158200600, 11655000, 17990000, 7700000,
              13900000, 8550000, 15300000, 9000000, 11000000, 30600000, 28900000, 29000000,
              24600000, 209000000)

# Exceedance frequency curve corresponding to the severities above

scen.efc <- c( 1.00, 1.50, 0.70, 1.20, 0.20, 1.30, 0.80, 1.80, 1.10, 1.70, 0.90, 1.60, 1.40,
               0.30, 0.50, 0.40, 0.60, 0.10 )

# Exceedance frequency curves comparison

plot(sev.total, efc.IE1model(sev.total, lambda), xlim = c(0, 5e8),
     main = "Comparison of excedance frequency curves", ylim = c (0, 2.3), xlab = "severity",
     ylab = "efc", lty = 1, cex.main = 0.9, font.main = 1)
points(scen.sev, scen.efc, col = "blue")

legend("topright", c("Modelled efc" ,"Scenarios efc"), lty = c(1, 4), cex = 1.0,
        col = c("black",  "blue"))

# Exceedance frequency curves comparison: log-log plot

plot(log(sev.total), log(efc.IE1model(sev.total, lambda)), main = "Comparison of exceedance
     frequency curves (log scale)", cex = 0.2, xlab = "severity", ylab = "efc",
     cex.main = 0.9, font.main = 1)

points(log(scen.sev), log(scen.efc), col = "blue")
legend("topright", c("Modelled efc" ,"Scenarios efc"), lty = c(1, 4), cex = 0.9,
       col = c("black",  "blue"))

#----------------------Printing the table--------------------------------------------------

options (digits = 5)
table.efc <- cbind(sort(scen.sev), sort(scen.efc, decreasing = TRUE),
                                   efc.IE1model(sort(scen.sev), lambda))

colnames(table.efc) <- c("Scenario severity", "Scenario efc", "IE1 model efc")

print(kable(table.efc))
```