```
# Supplementary Data 1. R code to analyse diversity-multifunctionality relationships in
theoretical communities.

#####################################################################################
#                                                                                   #
# Investigate the diversity-multifunctionality relationship with theoretical data. #
#                                                                                   #
#####################################################################################

library(MASS)
library(devEMF)

setwd('fil-in-working-directory-path')

# richness_values
no_communities <- 100
richness_values <- rep(c(1:5),no_communities/5)
average_correlation_value <- c(-0.07,0,0.25,0.5,0.75,1)
regional_richness <- 20
no_functions <- 15
repetitions_artificial <- 100
function_variability <- 1 # standard deviation EF values
interaction_strength <- 0
EF_mean <- 0

filename <- paste("multifunctionality_artificial_data3_", ".emf", sep="")
emf(file = filename,width=15,height=20)
par(mfrow=c(3,2))

feta_each <- c()
for(a in 1:length(average_correlation_value)){
 effect_size_all <- c()
 lower_limit_all <- c()
 upper_limit_all <- c()
 p_value_all <- c()
 feta_all <- c()
 for(b in 1:repetitions_artificial){
  # calculate ecosystem values
  ecosystem_function <- matrix(ncol=no_functions,nrow=no_communities)
  if(function_variability>0){
   Sigma2=diag(no_functions)
   Sigma2[which(diag(no_functions)==0)]<-average_correlation_value[a]
   function_values_pool <- mvrnorm(n = regional_richness,
mu=rep(EF_mean,no_functions),Sigma=Sigma2,tol=1e-6,empirical=FALSE,EISPACK=FALSE)
   for(i in 1:no_communities){
    function_values <-
function_values_pool[sample(c(1:regional_richness),richness_values[i]),]
    interaction_values <- interaction_strength*sample(abs(function_values),
(richness_values[i]*richness_values[i]),replace=T)
    if(richness_values[i]==1){interaction_values<-0}
    if(richness_values[i]==1){
     for(j in 1:no_functions){
      ecosystem_function[i,j] <- function_values[j]
     }
    }else{
     for(j in 1:no_functions){
      ecosystem_function[i,j] <- mean(function_values[,j])+sum(interaction_values)
     }
    }
   }
  }else{
   for(i in 1:no_communities){
    for(j in 1:no_functions){
     ecosystem_function[i,j] <- 0
    }
   }
  }

  ecosystem_function2 <- ecosystem_function
  for(i in 1:no_communities){
   for(j in 1:no_functions){
    ecosystem_function2[i,j] <- (ecosystem_function[i,j] - min(ecosystem_function[,j]))
/
   (max(ecosystem_function[,j]) - min(ecosystem_function[,j]))
   }
  }

  ecosystem_function <- ecosystem_function2

  teller <- var(rowSums(function_values_pool))
  sd_values <- c()
  for(i in 1:no_functions){
   sd_values[i] <- sd(function_values_pool[,i])
  }
  noemer <- (sum(sd_values))^2
  feta_all[b] <- teller / noemer

  # calculate multifunctionality
  multifunctionality <- matrix(ncol=100,nrow=no_communities)
  for(k in 1:100){
   for(i in 1:no_communities){
    performance <- c()
    for(j in 1:no_functions){
     threshold <- min(ecosystem_function[,j])+(k/100)*(max(ecosystem_function[,j])-
min(ecosystem_function[,j]))
     if(ecosystem_function[i,j]>threshold){performance[j]<-1}else{performance[j]<-0}
    }
    multifunctionality[i,k] <- sum(performance)
   }
  }
  multifunctionality <- data.frame(multifunctionality)
  column_names <- c()
  for(i in 1:100){
   column_names[i] <- paste("multifunctionality",i,sep="")
  }
  names(multifunctionality) <- column_names
  row_names <- c()
  for(i in 1:no_communities){
   row_names[i] <- paste("community_",i,sep="")
  }
  row.names(multifunctionality) <- row_names

  multifunctionality_data <- cbind(richness_values,multifunctionality)

  # test richness effect
  p_value <- c()
  effect_size <- c()
  upper_limit <- c()
  lower_limit <- c()
  intercept <- c()
  for(i in 1:99){
   j = i + 1
   multifunctionality_data2 <-
cbind(multifunctionality_data,"focal_multifunctionality"=multifunctionality_data[,j])
   m1 <- lm(focal_multifunctionality~richness_values,data=multifunctionality_data2)
   output <- summary(m1)
   lower_limit[i] <- confint(m1, 2, level = 0.95)[1]
   upper_limit[i] <- confint(m1, 2, level = 0.95)[2]
   p_value[i] <- output$coefficients[2,4]
   effect_size[i] <- output$coefficients[2,1]
   intercept[i] <- output$coefficients[1,1]
  }
  effect_size_all <- cbind(effect_size_all,effect_size)
  lower_limit_all <- cbind(lower_limit_all,lower_limit)
  upper_limit_all <- cbind(upper_limit_all,upper_limit)
  p_value_all <- cbind(p_value_all,p_value)
 }
 effect_size_mean <- c()
 lower_limit_mean <- c()
 upper_limit_mean <- c()
 p_value_mean <- c()
 for(i in 1:99){
  effect_size_mean[i] <- mean(effect_size_all[i,])
  lower_limit_mean[i] <- mean(lower_limit_all[i,])
  upper_limit_mean[i] <- mean(upper_limit_all[i,])
  p_value_mean[i] <- mean(p_value_all[i,])
 }
 print(a)
 print(which(p_value_mean<0.05))
 #effect_size_mean
 print(which(effect_size_mean==max(effect_size_mean)))
 print(max(effect_size_mean))
 print(max(effect_size_mean)/(15/4))
 print(which(effect_size_mean==min(effect_size_mean)))
 print(min(effect_size_mean))
 print(min(effect_size_mean)/(15/4))

 feta_each[a] <- mean(feta_all)

 plot(effect_size_mean~c(1:99),xlab="threshold value (%)",ylab="change in number of
functions per added species",
 cex.axis=1.6,cex.lab=1.6,pch=19,xlim=c(1,100),ylim=c(-1.5,1.5))
 lines(c(1:99), upper_limit_mean, col = 'grey')
 lines(c(1:99), lower_limit_mean, col = 'grey')
 polygon(c(c(1:99), rev(c(1:99))), c(upper_limit_mean, rev(lower_limit_mean)),col =
"grey", border = NA)
 points(effect_size_mean~c(1:99),pch=19)
 lines(x=c(-10,1000),y=c(0.0,0.0),col="black",lty=2)
}
dev.off()

feta_each

###############################################################################
#                                                                             #
#                              End of script                                  #
#                                                                             #
###############################################################################
```