AutoAff

Yi Lian

2019-07-21

# set.seed(29010719)  
# names <- c(LETTERS, letters)  
# n <- length(names)  
# aff <- sapply(names, function(x){paste("U", x, sep = "")})  
#   
# a <- as.character(sample(aff, n, replace = TRUE))  
# b <- as.character(c(sample(aff, n - 8, replace = TRUE), rep(NA, 8)))  
# c <- as.character(c(sample(aff, n - 20, replace = TRUE), rep(NA, 20)))  
#   
# data <- data.frame(names, a, b, c)  
#   
#   
# for (i in 1:32) {  
# if (data[i,1] == data[i,2] | data[i,1] == data[i,3] | data[i,2] == data[i,3]) {  
# print(i)  
# }  
# }  
#   
# for (i in 1:44) {  
# if (data[i,1] == data[i,2]) {  
# print(i)  
# }  
# }  
#   
# data[11, 3] = "UZ"  
# data <- data[sample(1:n), ]  
  
data <- read.csv("Book1.csv", na.strings = c("", NA))  
  
AutoAff <- function(X, aff1, aff2, aff3, name) {  
 mat <- as.matrix(X[, c(aff1, aff2, aff3)])  
 aff <- na.exclude(unique(as.character(as.vector(t(mat)))))  
 n.aff <- length(aff)  
 aff <- aff[1:n.aff]  
   
 mat <- cbind(mat, rep(NA, dim(X)[1]))  
 for (i in 1:dim(X)[1]) {  
 aa <- which(aff == mat[i, 1])  
 bb <- which(aff == mat[i, 2])  
 cc <- which(aff == mat[i, 3])  
 if (length(bb) != 0){  
 if (length(cc) == 0) {  
 mat[i, 1] <- aff[min(c(aa, bb))]  
 mat[i, 2] <- aff[max(c(aa, bb))]  
 mat[i, 4] <- paste(min(c(aa, bb)), ",", max(c(aa, bb)), sep = "")  
 }  
 if (length(cc) != 0) {  
 order <- sort(c(aa, bb, cc))  
 mat[i, 1] <- aff[order[1]]  
 mat[i, 2] <- aff[order[2]]  
 mat[i, 3] <- aff[order[3]]  
 if (order[2] - order[1] == 1 & order[3] - order[2] == 1) {  
 mat[i, 4] <- paste(order[1], "-", order[3], sep = "")  
 } else {  
 mat[i, 4] <- paste(order[1],",", order[2], ",", order[3], sep = "")  
 }  
 }  
 }  
   
 }  
   
 data[, "a"] <- mat[, 1]  
 data[, "b"] <- mat[, 2]  
 data[, "c"] <- mat[, 3]  
 data[, "names"] <- data$LAST.NAME  
 data[, "index"] <- mat[, 4]  
   
 return(list(affilliations = aff, count = n.aff, data = data))  
}  
  
result <- AutoAff(data, "AFFILIATION.1", "AFFILIATION.2", "AFFILIATION.3", "LAST.NAME")  
data <- result$data

Wu1,2,, Levis3,4,, Sun1,,, Krishnan1,,, He1,4,, Riehm1,5,, Rice1,6,, Azar1,4,, Yan1,,, Neupane1,4,, Bhandari1,4,, Imran1,,, Sanchez1,,, Chiovitti1,,, Saadat1,,, Boruff7,,, Cuijpers8,,, Gilbody9,,, McMillan9,,, Ioannidis10,,, Kloda11,,, Patten12,13,14, Shrier3,4,, Ziegelstein15,,, Henry1,,, Ismail16,17,18, Loiselle19,20,, Michell21,22,, Tonelli23,,, Al-Adawi24,,, Beraldi25,,, Braeken,,, Büel-Drabe26,,, Bunevicius27,28,, Carter29,30,, Chen,,, Cheung31,,, Clover,,, Conroy32,,, Costa-Requena,,, Cukor33,,, da Rocha e Silva,,, Dabscheck34,35,, Daray36,37,, De Souza38,39,, Douven,,, Downing40,41,, Feinstein42,43,, Ferentinos44,45,, Fischer46,,, Flint47,48,, Fujimori49,,, Gallagher50,,, Gandy51,,, Goebel,,, Grassi52,53,, Härter54,,, Jenewein55,56,, Jetté57,,, Julião58,,, Keller,,, Kim59,,, Kim60,,, Kjærgaard61,62,, Köhler63,,, Loosman64,,, Love65,,, Löwe66,,, Martin-Santos67,68,, Massardo69,,, Matsuoka70,71,, Mehnert ,,, Michopoulos72,,, Misery73,,, Navines67,74,, O’Donnell75,76,, O’Rourke77,,, Öztürk78,,, Peceliuniene79,,, Pintor80,81,82, Ponsford83,84,, Quinn85,,, Reme86,87,, Reuter88,,, Rooney89,90,, Sánchez-González91,92,93, Schwarzbold94,,, Senturk Cankorur95,,, Shaaban,,, Sharpe96,,, Sharpe,,, Simard97,98,99, Singer100,,, Stafford101,102,, Stone103,,, Sultan104,105,, Teixeira106,107,, Tiringer108,,, Tung,,, Turner109,110,111, Walker112,,, Walterfang113,114,115, Wang116,,, White117,,, Wong40,118,, NA,,, Benedetti4,119,120, Thombs,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,, NA,,.