

Negotiation Exercise Paring Script

Daisung Jang

Thursday, July 7, 2015

Contents

1	Overview and Instructions	1
2	Exercise 1 Pairings	5
3	Exercise 2 Pairings	6
4	Exercise 3 Pairings	7

1 Overview and Instructions

The purpose of this script is to generate negotiation pairings for three exercises. Pairings for more exercises is possible given slight alterations to the script.

Please alter the list of names/input your own CSV file of name before running this script.

Pairings generated conform to the following rules.

- People will not negotiate with the same person twice

If there is an odd number of people, the script will automatically add a TA (currently “TA”) to the list of people, then create randomizations. The name of the TA can be altered in the script.

Repeated execution of this script will produce the same pairings, given the same list of people - the random assignment procedure uses “set.seed()” call. Alter/comment this line to ensure different pairings with each execution of the script.

This script is free to use for research and teaching purposes. You are free to modify and improve the script as you see fit.

```
#read in the names data - you can substitute a CSV file instead
name <- c("Person1", "Person2", "Person3", "Person4", "Person5", "Person6", "Person7")

class <- as.data.frame(name)

#show the example dataset
class
```

```
##      name
## 1 Person1
## 2 Person2
## 3 Person3
## 4 Person4
## 5 Person5
## 6 Person6
## 7 Person7
```

```

#change class of variables, and create an ID variable
class$name <- as.character(class$name)

#check if dataset contains even or odd number of people
if(((length(class$name) %% 2) == 0) == FALSE) { class[length(class$name)+1, "name"] <- "TA" }

#check the sample data to see if "TA" has been inserted
class

```

```

##      name
## 1 Person1
## 2 Person2
## 3 Person3
## 4 Person4
## 5 Person5
## 6 Person6
## 7 Person7
## 8      TA

```

```

x <- class

y1 <- class

y2 <- class

y3 <- class

x$pairing1 <- "empty"

x$pairing2 <- "empty"

x$pairing3 <- "empty"

x$ex1_pair_num <- NA

x$ex2_pair_num <- NA

x$ex3_pair_num <- NA

#counters for pair numbers
p1 = 0
p2 = 0
p3 = 0

#First Loop - generates the first set of pairings.
for(i in 1:length(x$name)){

  set.seed(111)

  pair1 <- y1[sample(nrow(y1), 1), ]

  empty <- ifelse(x[i, c("pairing1")] == "empty", 1, 0)

```

```

x[i, ]$pairing1 <- ifelse(empty == 1, pair1, x[i, ]$pairing1)

z <- as.numeric(rownames(subset(x, name == pair1)))

x[z, c("pairing1")] <-
  ifelse(x[z, c("pairing1")] == "empty" & empty == 1, x[i, ]$name, x[z, c("pairing1")])

if(length(z) >= 1 & empty == 1)
  {y1 <- subset(y1, name != pair1 & name != x[z, c("pairing1")])}

if(empty == 1) { p1 <- p1 +1 }

if(empty == 1) { x[i, "ex1_pair_num"] <- p1 }

}

#Second Loop - generates the second set of pairings.
for(i in 1:length(x$name)){

  y2_1 <- subset(y2, name != x[i, ]$pairing1)

  set.seed(111)

  pair2 <- y2_1[sample(nrow(y2_1), 1), ]

  empty <- ifelse(x[i, c("pairing2")] == "empty", 1, 0)

  x[i, ]$pairing2 <- ifelse(empty == 1, pair2, x[i, ]$pairing2)

  z <- as.numeric(rownames(subset(x, name == pair2)))

  x[z, c("pairing2")] <-
    ifelse(x[z, c("pairing2")] == "empty" & empty == 1, x[i, ]$name, x[z, c("pairing2")])

  if(length(z) >= 1 & empty == 1)
    {y2 <- subset(y2, name != pair2 & name != x[z, c("pairing2")])}

  if(empty == 1) { p2 <- p2 +1 }

  if(empty == 1) { x[i, "ex2_pair_num"] <- p2 }

}

#Third Loop - generates the third set of pairings.
for(i in 1:length(x$name)){

  y3_1 <- subset(y3, name != x[i, ]$pairing2 & name != x[i, ]$pairing1)

  set.seed(111)

  pair3 <- y3_1[sample(nrow(y3_1), 1), ]

  empty <- ifelse(x[i, c("pairing3")] == "empty", 1, 0)

```

```

x[i, ]$pairing3 <- ifelse(empty == 1, pair3, x[i, ]$pairing3)

z <- as.numeric(rownames(subset(x, name == pair3)))

x[z, c("pairing3")] <-
  ifelse(x[z, c("pairing3")] == "empty" & empty == 1, x[i, ]$name, x[z, c("pairing3")])

if(length(z) >= 1 & empty == 1)
  {y3 <- subset(y3, name != pair3 & name != x[z, c("pairing3")])}

if(empty == 1) { p3 <- p3 + 1 }

if(empty == 1) { x[i, "ex3_pair_num"] <- p3 }
}

library(xtable)

ex1 <- subset(x, ex1_pair_num > 0)

ex1 <- as.data.frame(cbind(ex1$name, ex1$pairing1))

names(ex1)[names(ex1) == 'V1'] <- 'Role 1'

names(ex1)[names(ex1) == 'V2'] <- 'Role 2'

ex2 <- subset(x, ex2_pair_num > 0)

ex2 <- as.data.frame(cbind(ex2$name, ex2$pairing2))

names(ex2)[names(ex2) == 'V1'] <- 'Role 1'

names(ex2)[names(ex2) == 'V2'] <- 'Role 2'

ex3 <- subset(x, ex3_pair_num > 0)

ex3 <- as.data.frame(cbind(ex3$name, ex3$pairing3))

names(ex3)[names(ex3) == 'V1'] <- 'Role 1'

names(ex3)[names(ex3) == 'V2'] <- 'Role 2'

```

2 Exercise 1 Pairings

```
print(xtable(ex1), size="\\Large", comment=FALSE)
```

	Role 1	Role 2
1	Person1	Person5
2	Person2	Person6
3	Person3	Person7
4	Person4	TA

3 Exercise 2 Pairings

```
print(xtable(ex2), size="\\Large", comment=FALSE)
```

	Role 1	Role 2
1	Person1	Person6
2	Person2	Person5
3	Person3	Person4
4	Person7	TA

4 Exercise 3 Pairings

```
print(xtable(ex3), size="\\Large", comment=FALSE)
```

	Role 1	Role 2
1	Person1	Person4
2	Person2	Person7
3	Person3	Person6
4	Person5	TA