

Procédure de transformation d'une matrice de flux fermée non carrée en une matrice fermée et carrée

F. Bahoken

Janvier, 2016

1. Présentation :

Cette annexe décrit une procédure permettant de transformer une matrice de flux non carrée (n.m), formée de n lignes et de m colonnes d'effectifs différents, en une matrice carrée où le nombre de lignes est égal au nombre de colonnes (n.n). On utilise pour cela une liste (n) d'identifiants contenant l'ensemble des lieux de la zone d'étude. Le cas d'illustration est la matrice de flux domicile-travail de l'Isère qui n'est pas carrée.

Données :

source.csv matrice de flux de l'Isère (non carrée, au format liste) **CODE_Isere** : liste des identifiants (code_insee) des communes

2. Préparation

Connection et déclaration des packages

```
#purge environnement de travail
rm(list=ls())
setwd("C:/FRANCOISE/R/4-Carree")

# declaration des packages requis
library(reshape2)
library(rgdal)
```

Chargement de la liste des identifiants

```
tab <- read.csv2("CODE_Isere.csv",header=TRUE,sep=";",stringsAsFactors=FALSE,encoding="UTF-8",dec=".",check.na
mes=FALSE)
str(tab)
```

```
## 'data.frame': 533 obs. of 1 variable:
## $ CODE: int 38001 38002 38003 38004 38005 38006 38008 38009 38010 38011 ...
```

3. Construction d'une matrice vide et transformation de la matrice non carrée en matrice carrée (format matriciel)

Construction d'une matrice vide

```
# Construction d'une matrice (mat) à 0, de dimension (n*n) où n est le nombre de lignes correspondant au nombre
d'identifiants

nbi<-dim(tab)[1]
mat <- matrix(data = c(0), nrow = nbi, ncol = nbi,dimnames=list(c(as.matrix(tab)),c(as.matrix(tab))))
class(mat)
```

```
## [1] "matrix"
```

Transformation de cette matrice au format liste

```
#Transformation de cette matrice en une liste (3 colonnes) et création d'un nouvel identifiant unique pour cha
que couple de lieu (ID_link), issu de la concaténation des codes des lieux d'origine et de destination

tabflow <-melt(mat)
names(tabflow) = c("j", "i","Fij")
tabflow$ID_link<-paste(tabflow$i,tabflow$j,sep = "-")
head(tabflow)
```

```
##      j      i Fij      ID_link
## 1 38001 38001  0 38001-38001
## 2 38002 38001  0 38001-38002
## 3 38003 38001  0 38001-38003
## 4 38004 38001  0 38001-38004
## 5 38005 38001  0 38001-38005
## 6 38006 38001  0 38001-38006
```

Chargement des données de flux observées (matrice non carrée au format liste)

```
#Chargement de la liste des valeurs de flux observées (data), dont on sait qu'elle n'est pas carrée.
#Création d'un nouvel identifiant unique pour chaque couple de lieu (ID_link), comme précédemment. Le lieu d'o
rigine =COMMUNE et le lieu de destination=DCLT, la variable COUNT correspond à la valeur du flux.

data <- read.csv2("source.csv",header=TRUE,sep=";",stringsAsFactors=FALSE,encoding="UTF-8",dec=".",check.names
=FALSE)
data$ID_link<-paste(data$COMMUNE,data$DCLT,sep = "-")
head(data)
```

```
##  COMMUNE  DCLT    COUNT    ID_link
## 1   38001 38001 527.7349 38001-38001
## 2   38001 38012  15.3317 38001-38012
## 3   38001 38013   7.6658 38001-38013
## 4   38001 38022  30.6634 38001-38022
## 5   38001 38028   3.8329 38001-38028
## 6   38001 38029  11.4988 38001-38029
```

Mise en relation des Id_link des deux listes et construction de la matrice Fij

```
tabflow= data.frame(tabflow, data[match(tabflow[, "ID_link"], data[, "ID_link"]),2:3])
Fij<-data.frame(tabflow$i,tabflow$j,tabflow$COUNT)
names(Fij) = c("i", "j", "COUNT")
head(Fij)
```

```
##      i      j    COUNT
## 1 38001 38001 527.7349
## 2 38001 38002      NA
## 3 38001 38003      NA
## 4 38001 38004      NA
## 5 38001 38005      NA
## 6 38001 38006      NA
```

Transformation de la liste Fij en matrice carrée (matflow)

```
matflow<-acast(Fij,i~j)
```

```
## Using COUNT as value column: use value.var to override.
```

```
dim(matflow)
```

```
## [1] 533 533
```

```
diag(matflow)<-0
```

Suppression des NA

```
#Les cases vides de cette matrice étant renseignées par NA, on les met à zéro.
for (i in 1:nrow(matflow))
  for (j in 1:ncol(matflow))
    if (is.na.data.frame(matflow[i,j])==TRUE) {matflow[i,j]<-0}

#head(matflow)
```

Export de la matflow en (Fij), pour cartographie et/ou traitements ultérieurs

```
write.table(Fij,file="./matflow.csv",sep=";",eol="\n")
```