Memoria de Cálculo SIBIO

Renato Vargas, Andrés Mondaini

2022-11-07

## Preámbulo

En esta memoria de cálculo se muestra de manera anotada el código utilizado para procesar los cuadros de oferta y utilización para convertirlos de cuadros de entrada múltiple a archivos planos estructurados.

Es importante extender las habilidades de R con ciertas librerías las cuales se llaman al inicio:

library(readxl)  
library(openxlsx)  
library(reshape2)  
library(stringr)  
library(plyr)

Se limpia el área de trabajo y nos aseguramos de colocarnos en el directorio en el que se encuentran los datos. Además, ponemos el nombre del archivo en una variable a la que podamos hacer referencia sin necesidad de escribirlo completo más adelante. Por ejemplo, para el caso de Colombia.

rm(list = ls())  
wd <- "C:/datos/COL"  
setwd(wd)  
archivo <- "COU\_2014-2019\_PRECIOSCORRIENTES\_66x61\_REFERENCIA.xlsx"

En este punto es necesario aclarar que el archivo de Excel debe cumplir con ciertos criterios para poder extraer y procesar la información programáticamente. En esta guía no se muestra la recursión que permite acceder y procesar todos los años de una sola vez. Invitamos al lector a inspeccionar el código fuente para esto, pero basta con saber que el bucle recursivo itera por todas las pestañas del archivo de Excel, ejecutando el código que aquí se explica para cada año.

1. Verificación que todos los cuadros de un año base sean iguales.
2. Se debe contar con un set de cuadros de oferta utilización, valor agregado y empleo si lo hubiere por pestaña[[1]](#footnote-20).
3. Debe existir en cada pestaña celdas específicas que determinen el año y el tipo de precios (corrientes o constantes). Estos deben estar siempre en las mismas celdas.

Habiendo cumplido con estos requrimientos, primeramente obtenemos un listado de las pestañas con las que cuenta nuestro archivo de Excel y las colocamos en una variable que denominamos hojas.

hojas <- excel\_sheets(archivo)

A través de índices, se puede acceder a cualquier objeto de la lista según su posición. Por ejemplo, hojas[4] hace referencia al cuarto objeto de la lista de pestañas cuyo valor es "2015". Al anteceder cualquier posición con el signo menos (-), negamos el índice, por lo que hojas[4] equivale decir “muéstrame todos los valores de la lista, excepto el cuarto”. De esta manera, podemos reducir nuestro listado para concentrarnos solamente en los que contienen los cuadros que queremos procesar negando, no solamente una posición, sino un listado de posiciones usando el símbolo (-) y el concatenador de listas c().

hojas <- hojas[-c(1,3,5,7,9,11,13,14,15,16,  
 17,18,19,20,21,22,23,24,25)]

Extraemos el año y la unidad de medida del archivo de excel obteniendo los datos de un grupo de celdas que incluye el año y el tipo de precios, los cuales colocamos en un marco de datos llamado info. Nótese que en este punto nos referimos a las hojas de Excel a través de nuestro listado hojas (2 en este caso). En la versión recursiva, el 2 se reemplaza por un iterador que cambia para cada vuelta del bucle, según las hojas de la lista. De ese objeto info extraemos el año y la unidad de medida (“millones de pesos”, por ejemplo) y los colocamos en las variables anio y unidad. En este análisis solamente trabajamos con precios corrientes, así que colocamos ese identificador en una variable llamada precios.

info <- read\_excel(  
 archivo,  
 range = paste("'", hojas[2], "'!a4:a5", sep = "") ,  
 col\_names = FALSE,  
 col\_types = "text",  
 )  
anio <- as.numeric((str\_extract(info[2, ], "\\d{4}")))  
unidad <- toString(info[1, ])   
precios <- "Corrientes"

## Cuadro de Oferta

En este punto leemos los datos crudos como matriz del cuadro de oferta y los colocamos en una variable denominada oferta. Únicamente importamos valores, omitimos las celdas con nombres de columna o de fila.

oferta <- as.matrix(read\_excel(  
 archivo,  
 range = paste("'" , hojas[i], "'!c11:cb78", sep = ""),  
 # Nótese que no incluimos la fila de totales  
 col\_names = FALSE,  
 col\_types = "numeric"  
 ))

Las siguientes dos líneas asignan un correlativo a cada fila y a cada columna, respectivamente ("of001", "of002, etc.) en donde “of” significa “oferta filas” y ("oc001", "oc002", etc.) en donde “oc” significa “oferta columnas”. Estos correlativos nos permitirán identificar según nuestro cuadro de equivalencias a qué transacción y actividad económica se refieren las columnas y a qué transacción y producto se refieren las filas.

rownames(oferta) <- c(sprintf("of%03d", seq(1, dim(oferta)[1])))  
colnames(oferta) <- c(sprintf("oc%03d", seq(1, dim(oferta)[2])))

Una de las particularidades de la manera en que las instituciones encargadas elaboran los cuadros de oferta y utilización es que incluyen muchas veces filas y columnas con sumas de subtotales, así como filas y columnas vacías. En nuestra base de datos es necesario contar únicamente con datos básicos que no sean duplicados. Por esa razón utilizamos la técnica de índices mostrada anteriormente para eliminar filas y columnas y hacer más compactos los cuadros con solo lo esencial. Este punto es importante hacerlo únicamente después de haber puesto los números correlativos a las filas y columnas, pues queremos saber exactamente a qué concepto se refiere cada celda, según nuestro cuadro de equivalencias.

La diferencia con el índice mostrado anteriormente es que en el primer caso referente a una lista, el índice es un valor único que denota la posición del elemento en la lista al que hacemos referencia. En una matriz de datos, como la que hemos importado del cuadro de oferta, cada posición está dada por dos valores separados por una coma, como en la notación matricial. El primero denota el número de fila en el que se encuentra la celda a la que hacemos referencia y el segunddo denota el número de columna. De igual manera a como nos referimos a varias hojas en la lista de las pestañas de Excel a través de una lista de números aquí podemos referirnos a todas las columnas y filas que no necesitamos pasando listas en vez de valores únicos a cada una de las posiciones del índice. Con el negamos el listado, obteniendo el inverso o en este caso, solamente las filas y columnas que necesitamos para nuestro análisis.

En el caso de Colombia, no hay filas vacías y por esa razón dejamos vacío el espacio antes de la coma, para que R tome en cuenta todas las filas. El listado después de la coma y el signo (-) son todas las columnas vacías o con cálculos de subtotales que no necesitamos.

oferta <- oferta[ , -c(1,2,3,4,5,6,7,8,71,72,73,74,75,76)]

El siguiente paso es el más importante de este procesamiento, pues es como convertimos el cuadro de entrada doble por el archivo plano estructurado. Esto se hace a través de la función melt() de la librería reshape2. El resultado de operar ese comando sobre el objeto oferta dará como resultado un marco de datos de tres columnas. Una con el identificador correlativo de fila, una con el de columna y una con el valor.

melt(oferta)

Aprovechamos para embeber la función anterior dentro de otra a través de la cual uniremos ese resultado a través de las columnas con otros valores que nos ayudarán a distinguir este cuadro de oferta de los de otros años y precios que procesemos a través de la función cbind(). Y finalmente le ponemos unos nombres más descriptivos a las columnas resultantes a través de colnames(oferta).

oferta <- cbind(anio, precios,1, "Oferta", melt(oferta), unidad)  
colnames(oferta) <-  
 c("Año",  
 "Precios",  
 "No. Cuadro",  
 "Cuadro",  
 "Filas",  
 "Columnas",  
 "Valor",  
 "Unidades")

Seguidamente, procesamos de la misma manera el cuadro de utilización, el valor agregado y el cuadro de empleo si lo hubiere y luego de crear todos los archivos planos con la misma estructura, se unen a través de las filas uno sobre el otro hasta contar con un solo marco de datos para el año que contiene oferta, utilización, valor agregado y empleo.

## Cuadro de utilización, valor agregado y empleo

A continuación ponemos el código de la utilización, valor agregado y empleo sin comentarios pues simplemente replican lo ya explicado.

# Se leen los datos  
utilizacion <- as.matrix(read\_excel(  
 archivo,  
 range = paste("'" , hojas[i], "'!D169:DQ320", sep = ""),  
 col\_names = FALSE,  
 col\_types = "numeric"  
))  
  
# Nombres correlativos  
rownames(utilizacion) <-  
 c(sprintf("uf%03d", seq(1, dim(utilizacion)[1])))  
colnames(utilizacion) <-  
 c(sprintf("uc%03d", seq(1, dim(utilizacion)[2])))  
  
# Columnas a eliminar con subtotales y totales  
utilizacion <- utilizacion[,-c(93, 98, 108, 109, 112, 115, 118)]  
  
# Desdoblamos  
utilizacion <-  
 cbind(anio,  
 precios,  
 2,  
 "Utilización",  
 melt(utilizacion),  
 unidad)  
  
# Nombres de columna descriptivos  
colnames(utilizacion) <-  
 c("Año",  
 "Precios",  
 "No. Cuadro",  
 "Cuadro",  
 "Filas",  
 "Columnas",  
 "Valor",  
 "Unidades")

Se hace el mismo procedimiento con el valor agregado.

valorAgregado <- as.matrix(read\_excel(  
 archivo,  
 range = paste("'" , hojas[i], "'!D325:DH330", sep = ""),  
 col\_names = FALSE,  
 col\_types = "numeric"  
))  
  
rownames(valorAgregado) <-  
 c(sprintf("vf%03d", seq(1, dim(valorAgregado)[1])))  
colnames(valorAgregado) <-  
 c(sprintf("vc%03d", seq(1, dim(valorAgregado)[2])))  
  
valorAgregado <- valorAgregado[,-c(93, 98, 108, 109)]  
  
valorAgregado <-  
 cbind(anio,  
 precios,  
 3,  
 "Valor Agregado",  
 melt(valorAgregado),  
 unidad)  
  
colnames(valorAgregado) <-  
 c("Año",  
 "Precios",  
 "No. Cuadro",  
 "Cuadro",  
 "Filas",  
 "Columnas",  
 "Valor",  
 "Unidades")

Y finalmente con el empleo

empleo <- as.data.frame(read\_excel(  
 archivo,  
 range = paste("'" , hojas[i], "'!D332:DH332", sep = ""),  
 col\_names = FALSE,  
 col\_types = "numeric"  
))  
  
rownames(empleo) <- c(sprintf("ef%03d", seq(1, dim(empleo)[1])))  
colnames(empleo) <- c(sprintf("ec%03d", seq(1, dim(empleo)[2])))  
  
empleo <- empleo[,-c(93, 98, 108, 109)]  
  
empleo <- cbind(anio,  
 precios,  
 4,  
 "Empleo",  
 "ef001",  
 melt(empleo),  
 "Puestos de trabajo")  
  
colnames(empleo) <- c("Año",  
 "Precios",  
 "No. Cuadro",  
 "Cuadro",  
 "Filas",  
 "Columnas",  
 "Valor",  
 "Unidades")

## Todo el COU en formato plano estructurado

Con los archivos planos estructurados de cada cuadro preparados, dado que comparten las mismas columnas en el mismo orden, simplemente los unimos a través de las filas, uno sobre el otro. A partir del resultado, creamos un objeto con la union y lo nombramos "SCN".

SCN <- rbind(oferta,  
 utilizacion,  
 valorAgregado,  
 empleo)

Finalmente, importamos nuestras tablas de equivalencia para filas y columnas. Estas son pestañas en un Excel en el que, de acuerdo a los cuadros de oferta y utilización originales, determinamos con qué transacción, actividad económica y producto se asocia cada uno de los correlativos que creamos al principio, utilizando una columna para cada correspondencia, haciéndolo una sola vez.

clasificacionColumnas <- read\_xlsx(  
 "filas\_y\_columnas.xlsx",  
 sheet = "columnas",  
 col\_names = TRUE,  
)  
clasificacionFilas <- read\_xlsx(  
 "filas\_y\_columnas.xlsx",  
 sheet = "filas",  
 col\_names = TRUE,  
)

Con las tablas de equivalencia importadas, utilizamos la función join() de la librería plyr para vincular los correlativos creados con las transacciones, actividades económicas y productos repitiendo cuantas veces sea necesario cada elemento. Esto lo hacemos con los valores de las columnas “Filas” y “Columnas” los cuales tienen los correlativos en ambas tablas. Le llamamos SCN al objeto final creado.

SCN <- join(SCN,clasificacionColumnas,by = "Columnas")  
SCN <- join(SCN,clasificacionFilas, by = "Filas")

Y finalmente lo exportamos a Excel.

write.xlsx(  
 SCN,  
 "SCN\_BD.xlsx",  
 sheetName= "SCNGT\_BD",  
 rowNames=FALSE,  
 colnames=FALSE,  
 overwrite = TRUE,  
 asTable = FALSE  
)

.

1. Algunos países utilizan una pestaña distinta para cada cuadro. En ese caso se deberá copiar y pegar previamente todos los cuadros de un año en una sola pestaña asegurándose que para todos los años los cuadros están exactamente en las mismas celdas. [↑](#footnote-ref-20)