Memoria de Procesamiento COUs

Renato Vargas

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

# Procesamiento Cuadros de Oferta y Utilización - Años 2013-2020

## Introducción

Como primer paso llamamos las librerías necesarias para el procesamiento de datos.

library(readxl)  
library(openxlsx)  
library(reshape2)  
library(stringr)  
library(plyr)  
library(RSQLite)  
library(DBI)  
library(readr)

Limpiamos el área de trabajo para evitar problemas.

rm(list = ls())

Es importante conocer la estructura de archivos de nuestro análisis y es necesario mantener un mapa mental de los diferentes directorios en los que tenemos nuestros datos y scripts de procesamiento. En el caso de nuestro repositorio de análisis, nos interesa el directorio datos.

scn\_scae\_gt  
 ├───datos  
 ├───documentos  
 │ └───presentaciones  
 └───scripts

Empezamos por crear variables con datos básicos de nuestro análisis. El nombre del país que estamos analizando, su código ISO de tres letras y la ruta al archivo en donde se encuentran nuestros datos. Nótese que normalmente si utilizamos un proyecto de RStudio, nuestra raíz se encontrará en donde se encuentra el archivo .RProj. No obstante, los documentos de Markdown tienen como raíz, el directorio en el que se encuentra el archivo por lo que debemos hacer referencia a los datos tomando esto en consideración.

pais <- "Guatemala"  
iso3 <- "GTM"  
archivo <- "../../datos/GTM\_COUS\_DESAGREGADOS\_2013\_2020.xlsx"

Nuestro primer paso exploratorio será abrir el archivo de Excel y ver con qué pestañas cuenta. Colocamos el listado de los nombres de las pestañas en un vector llamado hojas.

hojas <- excel\_sheets(archivo)  
hojas

## [1] "2013 C" "2014 C" "2014 K" "2015 C" "2015 K" "2016 C" "2016 K" "2017 C"  
## [9] "2017 K" "2018 C" "2018 K" "2019 C" "2019 K" "2020 C" "2020 K"

La documentación del archivo de Excel nos indica que cada pestaña cuenta con los cuadros de oferta y utilización para el año al que hace referencia el nombre y estos pueden ser a precios corrientes si están identificados con la letra “C” o precios constantes con la letra “K”. El análisis de la cuenta de energía no utiliza los precios constantes. Por conveniencia de análisis eliminamos las hojas de este tipo de precios de nuestro listado a través de índices inversos pasando al índice [ ]un signo menos seguido de un vector con las posiciones a eliminar -c().

hojas <- hojas[-c(3,5,7,9,11,13,15)]  
hojas

## [1] "2013 C" "2014 C" "2015 C" "2016 C" "2017 C" "2018 C" "2019 C" "2020 C"

En el código a continuación se encuentran múltiples instancias de la construcción hojas[i]. En vez de utilizar un número absoluto (ej. hojas[2]) para hacer referencia a la posición en el listado de la pestaña a analizar, utilizamos i, para poder después reutilizar el código de manera recursiva para analizar todas las hojas de manera programática, cambiando i por la posición de la pestaña en cada iteración. No haremos esto en esta guía, pero es un buen paso preparatorio.

i <- 1

## Procesamiento inicial del archivo de Excel

Primeramente obtenemos información de referencia del archivo de Excel que hemos inspeccionado visualmente en esa aplicación anteriormente. Sabemos en qué celdas de cada cuadro se encuentra esa información. Extraemos un rectángulo de celdas (una matriz) que cuenta con lo que nos interesa.

info <- read\_excel(  
 archivo,  
 range = paste("'" , hojas[i], "'!A8:B9", sep = "") ,  
 col\_names = FALSE,  
 col\_types = "text")

## ...1 ...2   
## [1,] "AÑO:" "2013"  
## [2,] "Millones de quetzales" NA

Y vemos dos datos importantes que colocamos en dos variables. Utilizando la función str\_extract( ,“\d{4}”) que combina la expresión regular que indica que debe encontrar cuatro números juntos obtenemos el año. Y la unidad de medida la obtenemos de la celda en la segunda fila y primera columna.

anio <- as.numeric((str\_extract(info[1,2 ], "\\d{4}")))  
unidad <- toString(info[2,1 ]) # unidad de medida

## [1] 2013

## [1] "Millones de quetzales"

Creamos un bucle para establecer otras variables de apoyo según tengamos precios constantes o corrientes en nuestras variables iniciales.

# Determinación de precios corrientes o constantes  
if (unidad != "Millones de quetzales") {  
 precios <- "Constantes"  
 id\_precios <- 2  
 unidad <- c("Millones de quetzales en medidas encadenadas de volumen con año de referencia 2013")  
 id\_unidad <- 2  
}else {  
 precios <- "Corrientes"  
 id\_precios <- 1  
 id\_unidad <- 1 # Millones de quetzales  
}

## [1] 1

## [1] "Corrientes"

## [1] "Millones de quetzales"

El siguiente paso es empezar la ingesta de datos del cuadro de oferta. Aprovechamos para establecer el identificador único del cuadro de oferta:

id\_cuadro <- 1 # 1 Oferta monetaria

Como puede verse en el parámetro range de la función read\_excel(), lo que hacemos, en esencia, es leer todas las celdas numéricas del cuadro, incluyendo espacios vacíos, columnas y filas de subtotales, para poder clasificar cada celda respecto a cuatro dimensiones: la transacción a la que corresponde, la actividad económica que la realiza, el producto objeto de la transacción y una cuarta que captura los elementos no estándar comunes en los cuadros de oferta y utilización publicados por los bancos centrales e institutos de estadística a través de lo que llamamos “Áreas transaccionales de fila o de columna”.

Leemos todo el rectángulo que contiene datos.

oferta <- as.matrix(read\_excel(  
 archivo,  
 range = paste("'" , hojas[i], "'!C16:FL240", sep = ""),  
 col\_names = FALSE,  
 col\_types = "numeric"  
))

Y en este punto simplemente le damos una identificación correlativa a cada fila y a cada columna. En el caso de las filas, ambos cuadros de oferta y utilización, comparten exactamente los mismos valores. Por esa razón el correlativo solamente se compone de un identificador ISO3 de país (GTM en este caso), una letra “f” minúscula (de filas) y un correlativo de tres dígitos. Esto lo hacemos accediendo a los nombres de fila y columna con las funciones rownames() y colnames() y asignándoles la secuencia compuesta por los elementos descritos.

Las funciones esenciales para lograr esto son sprintf() combinada con ‘seq()’. Nótese que se embebe la función paste() que nos permite concatenar el texto del código de país (GTM por ejemplo) con el formato del correlativo de fila o columna. También embebemos la función dim() dentro de la función seq() la cual nos permite saber programáticamente las dimensiones de la matriz importada.

rownames(oferta) <- c(sprintf(paste(iso3,"f%03d", sep = ""),   
 seq(1, dim(oferta)[1])))  
colnames(oferta) <- c(sprintf(paste(iso3,"oc%03d", sep = ""),   
 seq(1, dim(oferta)[2])))  
# Y mostramos una parte para referencia  
oferta[c(1:5),c(1:5)]

## GTMoc001 GTMoc002 GTMoc003 GTMoc004 GTMoc005  
## GTMf001 5.346939 0.0000 0 0 0  
## GTMf002 4734.946097 0.0000 0 0 0  
## GTMf003 101.132353 0.0000 0 0 0  
## GTMf004 0.000000 659.4762 0 0 0  
## GTMf005 0.000000 1539.4168 0 0 0

El siguiente punto consiste en identificar las columnas a eliminar, las cuales contienen subtotales y totales, contando como 1 la primera celda leída en el módulo anterior. Esto lo hacemos inspeccionando visualmente el archivo de Excel en esa aplicación previamente. El resultado es:

129 SIFMI (Lo eliminamos porque no tiene datos en ningún año)  
130 P1 PRODUCCION (PB) SUBTOTAL DE MERCADO  
135 P1 PRODUCCION (PB) SUBTOTAL USO FINAL PROPIO  
147 P1 PRODUCCION (PB) SUBTOTAL NO DE MERCADO  
148 P1 TOTAL PRODUCCION (PB)  
149 VACIA  
150 VACIA  
153 P7 IMPORTACIONES (CIF) TOTAL  
155 TOTAL OFERTA (a precios básicos)  
160 D21 IMPUESTOS SOBRE PRODUCTOS TOTAL  
165 MARGENES DE DISTRIBUCION TOTAL  
166 TOTAL OFERTA (PC) TOTAL OFERTA (PC)

Seguidamnete, identificamos en nuestro Excel las filas vacías y que contienen subtotales para ser eliminadas. Estas consisten del listado:

c(214, 215, 216, 218, 219, 224, 225)

A través del concepto de índices [fila, columna], utilizado en la notación matricial para referirnos a la posición de una o más celdas en una matriz, a través del signo menos (-) eliminamos las filas y columnas identificadas en el paso anterior.

oferta <- oferta[-c(214, 215, 216, 218, 219,  
 224, 225),  
 -c(129, 130,135,147,148,149,  
 150,153,155,160,165,166)]

El resultado es una matriz rectangular de datos válidos y únicos. Es decir, sin subtotales y totales que duplican los datos básicos. El siguiente paso transforma esa matriz de dos dimensiones a una tabla de datos de una sola dimensión que nos va a permitir guardar y manipular nuestros datos a través de una base de datos y el lenguaje de consultas estructuradas SQL.

Esto se logra con la función melt() de la librería reshape2. Nótese que para ahorrar un paso creamos columnas adicionales que representan lo mínimo necesario para poder identificar el año, el identificador de a qué cuadro pertenecen los datos y la unidad de medida del cuadro.

Anteriormente, agregábamos un identificador de precios constantes o corrientes. No obstante, en aras de estandarizar este procedimiento para temas energéticos, ambientales de empleo y otros que no usan esa diferenciación, decidimos que la distinción de constantes o corrientes se hará a través del campo de unidad de medida id\_unidad. Es decir Millones de quetzales constantes.

Por el momento no se incluirá precios constantes.

# Unimos el resultado de melt() con columnas con los valores  
# de nuestras variables iniciales.  
oferta <- cbind(iso3, anio,id\_cuadro, melt(oferta), id\_unidad)  
  
# Y renombramos según nuestra nomenclatura y convenciones  
colnames(oferta) <-  
 c("iso3",  
 "anio",  
 "id\_cuadro",  
 "id\_fila",  
 "id\_columna",  
 "valor",  
 "id\_unidad")  
  
# Finalmente mostramos los primeros valores de la tabla  
# como referencia.  
head(oferta)

## iso3 anio id\_cuadro id\_fila id\_columna valor id\_unidad  
## 1 GTM 2013 1 GTMf001 GTMoc001 5.346939 1  
## 2 GTM 2013 1 GTMf002 GTMoc001 4734.946097 1  
## 3 GTM 2013 1 GTMf003 GTMoc001 101.132353 1  
## 4 GTM 2013 1 GTMf004 GTMoc001 0.000000 1  
## 5 GTM 2013 1 GTMf005 GTMoc001 0.000000 1  
## 6 GTM 2013 1 GTMf006 GTMoc001 0.000000 1