Actualización de la base de datos del MARCEG V2.0

Dante Ruiz

7 de diciembre de 2017

## Índice del documento

* Introducción
* Objetivo
* Fuentes de información
* Configuración de R
  + Librerías
  + Funciones
* Procesamiento y limpieza de datos
* Ganadería
  + Producción
  + Mano de obra
  + Capital
  + Tierra

## Introducción

El Modelo de Agentes Rurales en un Contexto de Equilibrio General (MARCEG)es una herramienta para analizar el impacto de políticas públicas económicas, sociales, ambientales y tecnológicas, así como de cambio climático en la economía rural. El modelo permite visualizar de manera simultánea el impacto de las políticas en variables como la producción, ingreso, gasto, producto interno bruto y cambios en el uso de suelo.

La unidad de estudio del MARCEG son los hogares rurales, los cuales pueden ser consumidores y/o productores. Así, el modelo plantea un problema de optimización donde el hogar maximiza su ingreso a partir de la maximización del beneficio (ganancias) de su producción. Posteriormente, el hogar maximiza su utilidad sujeto a su ingreso total máximo. Así el MARCEG, consiste en un problema de optimización no lineal en un sistema de ecuaciones que permiten caracterizar el comportamiento de los hogares rurales a nivel de entidad federativa, región, localidad y mercados de factores productivos (tierra, trabajo, capital), bienes y servicios.

Los parámetros de dichas ecuaciones se deben calibrar con información real para que puedan representar los patrones de la economía rural en México. Originalmente, cuando se diseñó el MARCEG V2.0 en 2015, la base de datos que se utilizó para calibrar el MARCEG fue la **Encuesta Nacional a Hogares Rurales (ENHRUM) de 2003** que desarrolló el Colegio de México (COLMEX).

La ENHRUM se diseñó con el objetivo de capturar características generales de los hogares rurales mexicanos como su ingreso, gasto, actividades productivas e información sobre migración. Las actividades productivas rurales consideradas son la agricultura, la ganadería, los recursos naturales y la de bienes y servicios. La unidad de estudio es el hogar y se construyó una muestra de 1,546 hogares en 80 localidades distribuidas en 24 zonas geográficas.

Considerando lo anterior, se busca actualizar la base de datos con la que se calibra el MARCEG utilizando la **Encuesta CONEVAL a hogares rurales de México 2013 (ENCHOR)**. La encuesta se realizó a una muestra representativa nacional 2,300 hogares en localidades de 500 a 2,499 habitantes. El propósito de la encuesta es determinar una línea base de la capacidad productiva de los hogares rurales del país, al inicio de la implementación de la Cruzada contra el Hambre. La estructura de la encuesta es similar a la de la ENHRUM y aplicó cuestionarios relacionados a producción de cultivos, ganadería, bienes y servicios, recursos naturales y otros ingresos y gasto.

## Objetivo

El propósito de este documento es documentar el procedimiento para procesar y limpiar las bases de datos originales de la Encuesta Coneval a Hogares Rurales 2013 (ENCHOR), para que se puedan utilizar en el MARCEG. De esta manera, se documentan observaciones a la base de datos, la descripción de los pasos y el código para obtener las tablas que requiere el modelo. Debido a que la base de datos maneja distintos temas sobre la economía rural, este documento utiliza el sector ganadero como ejemplo.

El software que se utiliza para lograr este propósito es R-Statistics, un paquete estadístico open source. Se recomienda instalar R-Studio en la computadora para ejecutar el código. Asimismo, es posible que sea necesario modificar la ruta de los archivos en caso de que estos llegarán a causar algún problema.

## Fuentes de información

La base de datos de la ENCHOR 2013 se puede descargar del portal [CONEVAL](http://www.coneval.org.mx/Informes/Evaluacion/ENCHOR_2013/ENCHOR_2013_DTA/ENCHOR_2013_DTA.zip). El archivo comprimido contiene una carpeta con la base de datos de hogares, una base de datos de localidades y catálogos. Es preciso mencionar que las tablas de las bases de datos vienen en formato STATA.

Para el propósito de este proyecto se utilizarán la base de hogares y los catálogos, los cuales tienen la siguiente estructura.

* **Base de Hogares\_DTA**
  + Activos
  + Ahorrocb
  + Ahorrorp
  + Ahorrosf
  + Alim
  + Bys
  + Contaculti
  + Containv
  + Contasol
  + Contasoltenen
  + Credifuente
  + Credifuenteh
  + Credigasto
  + Crediprestam
  + Cultian\_oi
  + Cultian\_pv
  + Cultipe
  + Eventosin
  + Otrosing\_bd
  + Otrosing\_ca
  + Otrosing\_oig
  + Otrosing\_prov
  + Otrosing\_ps
  + Otrosing\_serv
  + Parcelanph
  + Parcelaph
  + Parcelariego
  + Portada
  + Produccion\_ganade\_a
  + Produccion\_ganade\_ta
  + Productos\_ganade
  + Recursosnat
  + Resultadoh
  + Resultadoi
  + Seguros
  + Smaiz
  + Sociodem
  + Sueltas
  + Vivienda
* **Cuestionarios**
  + 1.PORTADA
  + 2.SOCIODEMOGRAFIA
  + 3.PARCELAS
  + 4.CULTIVOS ANUALES
  + 5.CULTIVOS PERMANENTES
  + 6.CONTABILIDAD DE CULTIVOS
  + 7.GANADERIA
  + 8.ACTIVOS
  + 9.BIENES Y SERVICIOS
  + 10.RECURSOS NATURALES
  + 11.VIVIENDA
  + 12.CRÉDITO
  + 13.AHORRO
  + 14.OTROS INGRESOS Y GASTOS
  + 15.EVENTOS INESPERADOS
  + 16.SEGUROS
  + 17.ALIMENTACIÓN
  + 18.RESULTADO DE LA ENTREVISTA

## Configuración de R

En esta sección se especifican las librerías que se requieren instalar y cargar en la distribución de R. Asimismo, se indican las funciones diseñadas ad-hoc para simplificar el código de este proyecto.

#### Librerías

Las siguientes librerías se utilizan para añadir funcionalidades especiales a la distribución base de R para facilitar el procesamiento y limpieza de bases de datos. Se tienen que instalar y cargar previo a ejecutar el código de este documento.

# Para instalar las liberías borrar el símbolo de # en las siguientes tres filas. Re introducir el símbolo # una vez instaladas.  
# install.packages("haven")  
# install.packages("dplyr")  
# install.packages("haven")  
# install.packages("tidyverse")  
  
# Cargar las librerías en el espacio de trabajo  
library(haven) # Librería para que R pueda leer bases de datos en STATA  
library(dplyr) # Librería con un sintaxis especial para manipular tablas (data frames)  
library(tidyr) # Librería con funciones para limpiar y estructurar tablas (data frames)  
library(tidyverse)

#### Funciones

Las siguientes funciones fueron definidas para simplificar tareas de procesamiento de las bases de datos.

* **procesaSTATA( “nombreArchivo” )**. Es una función que de la carpeta Base de Hogares\_DTA lee en R un archivo en formato STATA, lo carga en el espacio de trabajo, y dentro de la carpeta Bases\_Hogares\_CSV lo guarda en un archivo CSV para poder abrirlo en EXCEL. La función recibe un argumento en formato string, es decir entre " " con el nombre del archivo. Es preciso mencionar que no se tiene que indicar la extensión del archivo dta.

procesarSTATA <- function(nombreArchivo, ...){  
 # Area de oportunidad. Ver como en windows se puede abreviar el working directory para no tener que escribir toda la ruta.  
 inputStataDirectory <- "Base de Hogares\_DTA/"  
 tablaSTATA = read\_dta(paste(inputStataDirectory,nombreArchivo,".dta", sep = "", ...))  
 outputCSVDirectory <- "Bases\_Hogares\_CSV/"  
 write.csv(tablaSTATA, file = paste(outputCSVDirectory,nombreArchivo,".csv", sep = ""), row.names=FALSE)  
 return(tablaSTATA)   
 }

guardarTablaProcesada <- function(tabla){  
 nombreArchivo <- deparse(substitute(tabla))   
 outputTablasProcesadas <- "Bases\_Hogares\_Procesadas/"  
 write.csv(tabla, file = paste(outputTablasProcesadas,nombreArchivo,".csv", sep = ""), row.names=FALSE)  
 }

## Procesamiento y limpieza de datos

## Ganadería

En esta sección se describe el proceso para limpiar la ENCHOR correspondiente al sector ganadero y producir el input que necesita el MARCEG.

Los resultados de los cuestionarios correspondientes a la SECCION 7. GANADERIA de la ENCHOR 2013 están distribuidos en los siguientes **archivos**:

* Produccion\_ganade\_a.dta:
  + Esta base datos es sobre los animales que los hogares han tenido de noviembre de 2012 a octubre de 2013.
  + Cuadro 7A. Tipo de Animal
  + Cuadro 7B. Contabilidad
  + Cuadro 7C. Compras
  + Cuadro 7D. Ventas
* Produccion\_ganade\_ta.dta
  + Cuadro 7E. Alimentación de los animales
  + Cuadro 7F. Pastos e insumos
  + Cuadro 7G. Mano de obra
  + Cuadro 7H. Apoyo gubernamental
* Productos\_ganade.dta

Se hicieron los siguientes **catálogos** para clasificar campos en la base de datos:

* catalogo\_grupo\_animal.csv
* catalogo\_animales.csv
* catalogo\_estado.csv
* catalogo\_municipio.csv

inputCatGpoAnimal <- "Catalogos/catalogo\_grupo\_animal.csv"  
cat\_gpo\_animal <- read\_csv(inputCatGpoAnimal,   
 col\_types = cols( COD\_GPO = col\_character(),  
 NOMBRE\_GRUPO = col\_character()  
 ))  
  
inputCatAnimal <- "Catalogos/catalogo\_animales.csv"  
cat\_animal <- read\_csv(inputCatAnimal, col\_types = cols( COD\_ANIMAL = col\_character(),  
 NOMBRE\_ANIMAL = col\_character()  
 ))  
  
inputCatEstado <- "Catalogos/catalogo\_estado.csv"  
cat\_estado <- read.csv(inputCatEstado) %>%   
 mutate(CVE\_ENT = sprintf("%02d", CVE\_ENT))  
  
  
inputCatMunicipio <- "Catalogos/catalogo\_municipio.csv"  
cat\_municipio <- read.csv(inputCatMunicipio) %>%   
 mutate(  
 CVE\_ENT = sprintf("%02d", CVE\_ENT),  
 CVE\_MUN = sprintf("%03d", CVE\_MUN)  
 )

### Producción (Q)

Para obtener la producción del sector ganadero se utiliza la tabla de STATA **Produccion\_ganade\_a**. Esta tabla incluye las preguntas y respuestas de los cuadros siguientes:

* Cuadro 7A. Tipo de Animal
* Cuadro 7B. Contabilidad
* Cuadro 7C. Compras
* Cuadro 7D. Ventas

# Cargar base de datos en formato STATA a R y guardar base de datos en formato CSV.   
prod\_ganade\_a <- procesarSTATA("Produccion\_ganade\_a")  
  
# Desplegar el nombre de los campos  
names(prod\_ganade\_a)

## [1] "edo" "mun" "loc" "cons"   
## [5] "grupo" "s7c7ap01" "s7c7bp02" "s7c7bp03"   
## [9] "s7c7bp04" "s7c7bp05" "s7c7bp06" "s7c7bp07"   
## [13] "s7c7bp08" "s7c7bp09" "s7c7cp01" "s7c7cp02"   
## [17] "s7c7cp03a" "s7c7cp03b" "s7c7cp04" "s7c7cp05\_lc"   
## [21] "s7c7cp05\_lit" "s7c7cp06\_mc" "s7c7cp06\_mit" "s7c7cp07\_edo"  
## [25] "s7c7cp08" "s7c7dp01" "s7c7dp02" "s7c7dp03a"   
## [29] "s7c7dp03b" "s7c7dp04" "s7c7dp05" "s7c7dp06"   
## [33] "s7c7dp07\_lc" "s7c7dp07\_lit" "s7c7dp08\_mc" "s7c7dp08\_mit"  
## [37] "s7c7dp09\_edo" "s7c7dp10" "s7c7dp11" "estrato"   
## [41] "ponderador"

# Número de filas  
nrow(prod\_ganade\_a)

## [1] 40524

Se observa que la tabla de producción ganadera incluye 41 campos y 40,524 registros.

Ahora, se construye una tabla solo con los campos que se necesitan para sacar la producción de los hogares rurales. Asimismo, se hacen los JOINS correspondientes, para incluir de los catálogos: el nombre de la entidad federativa, municipio, grupo del animal y nombre del animal.

# Crear tabla ganaderia\_produccion  
ganaderia\_produccion <- prod\_ganade\_a %>%  
 select(cons, # número de cuestionario  
 edo, # clave estado  
 mun, # clave municipio  
 loc, # clave localidad  
 grupo, # clave grupo   
 s7c7dp01, # clave animal   
 s7c7dp02, # num animales vendidos en el periodo (0 si no vendieron)  
 s7c7dp03a, # precio de venta  
 s7c7dp03b, # 01 venta por cabeza o 02 venta total   
 s7c7dp05, # vendió dentro/fuera localidad  
 s7c7dp09\_edo, # clave estado de mpio y localidad  
 s7c7dp08\_mc, # clave municipio de la localidad  
 s7c7dp07\_lc, # clave localidad   
 s7c7dp11 # gasto en transporte para venta animales   
 ) %>%  
 mutate\_at(vars("cons","edo","mun", "loc", "grupo","s7c7dp01","s7c7dp03b", "s7c7dp05",  
 "s7c7dp07\_lc", "s7c7dp09\_edo", "s7c7dp08\_mc"), funs(as.character)) %>%  
 mutate\_at(vars("s7c7dp02", "s7c7dp03a","s7c7dp11"), funs(as.numeric)) %>%  
 mutate(cons = sprintf("%04d", as.numeric(cons)),  
 edo = sprintf("%02d", as.numeric(edo)),  
 mun = sprintf("%03d", as.numeric(mun)),  
 loc = sprintf("%04d", as.numeric(loc)))  
  
 # Para ajustar el formato hay que quitar NAs  
 #mutate(edo\_venta = sprintf("%02d", as.numeric(s7c7dp09\_edo))) %>%  
 #mutate(mun\_venta = sprintf("%03d", as.numeric(s7c7dp08\_mc))) %>%  
 #mutate(loc\_venta = sprintf("%04d", as.numeric(s7c7dp07\_lc)))  
  
  
# Hacer el join de catálogos  
ganaderia\_produccion <- left\_join(ganaderia\_produccion, cat\_estado, by = c("edo" = "CVE\_ENT"))  
ganaderia\_produccion <- left\_join(ganaderia\_produccion, cat\_municipio, by = c("edo" = "CVE\_ENT", "mun" = "CVE\_MUN"))  
ganaderia\_produccion <- left\_join(ganaderia\_produccion, cat\_gpo\_animal, by = c("grupo" = "COD\_GPO"))  
ganaderia\_produccion <- left\_join(ganaderia\_produccion, cat\_animal, by = c("s7c7dp01" = "COD\_ANIMAL"))  
  
# Mostar vista previa  
head(ganaderia\_produccion)

## # A tibble: 6 x 18  
## cons edo mun loc grupo s7c7d… s7c7… s7c7… s7c7… s7c7… s7c7… s7c7…  
## <chr> <chr> <chr> <chr> <chr> <chr> <dbl> <dbl> <chr> <chr> <chr> <chr>  
## 1 0001 15 051 0060 2 23 0 NA <NA> <NA> <NA> <NA>   
## 2 0001 15 051 0060 2 22 0 NA <NA> <NA> <NA> <NA>   
## 3 0001 15 051 0060 2 21 0 NA <NA> <NA> <NA> <NA>   
## 4 0001 15 051 0060 4 43 0 NA <NA> <NA> <NA> <NA>   
## 5 0001 15 051 0060 1 15 0 NA <NA> <NA> <NA> <NA>   
## 6 0001 15 051 0060 1 14 0 NA <NA> <NA> <NA> <NA>   
## # ... with 6 more variables: s7c7dp07\_lc <chr>, s7c7dp11 <dbl>, NOM\_ENT  
## # <fctr>, NOM\_MUN <fctr>, NOMBRE\_GRUPO <chr>, NOMBRE\_ANIMAL <chr>

La tabla anterior se encuentra desagregada por hogar y tipo de animal. Para obtener la información de la producción y el valor económico de ésta, se necesita calcular el monto total vendido. En la base de datos este se encuentra reportado por cabeza o monto total. En este sentido el siguiente código crea un campo con el monto total.

# Esta función calcula el monto total  
calcularMontoVentaTotal <- function(precioVenta, cantidadVenta, unidadDeVenta){  
 ifelse(unidadDeVenta == 1, precioVenta \* cantidadVenta,   
 ifelse(unidadDeVenta == 2, precioVenta,   
 ifelse(is.na(unidadDeVenta),0,0)))  
}  
  
  
# Calculo de la venta total  
ganaderia\_produccion <- ganaderia\_produccion %>%   
 mutate(montoVentaTotal = calcularMontoVentaTotal(  
 s7c7dp03a, # monto de venta (puede ser precio o monto total)  
 s7c7dp02, # num animales vendidos en el periodo  
 s7c7dp03b) # 01 venta por cabeza o 02 venta total  
 )  
  
# Mostrar vista previa  
head(ganaderia\_produccion)

## # A tibble: 6 x 19  
## cons edo mun loc grupo s7c7d… s7c7… s7c7… s7c7… s7c7… s7c7… s7c7…  
## <chr> <chr> <chr> <chr> <chr> <chr> <dbl> <dbl> <chr> <chr> <chr> <chr>  
## 1 0001 15 051 0060 2 23 0 NA <NA> <NA> <NA> <NA>   
## 2 0001 15 051 0060 2 22 0 NA <NA> <NA> <NA> <NA>   
## 3 0001 15 051 0060 2 21 0 NA <NA> <NA> <NA> <NA>   
## 4 0001 15 051 0060 4 43 0 NA <NA> <NA> <NA> <NA>   
## 5 0001 15 051 0060 1 15 0 NA <NA> <NA> <NA> <NA>   
## 6 0001 15 051 0060 1 14 0 NA <NA> <NA> <NA> <NA>   
## # ... with 7 more variables: s7c7dp07\_lc <chr>, s7c7dp11 <dbl>, NOM\_ENT  
## # <fctr>, NOM\_MUN <fctr>, NOMBRE\_GRUPO <chr>, NOMBRE\_ANIMAL <chr>,  
## # montoVentaTotal <dbl>

El siguiente código genera la tabla final de la producción del sector ganadero. La información presentada corresponde al número y monto de la venta de animales agrupado por hogar y tipo de animal.

valorProduccionGanadera <- ganaderia\_produccion %>%   
 group\_by(cons, edo, NOM\_ENT, mun, NOM\_MUN, grupo, NOMBRE\_GRUPO) %>%   
 # cantidad  
 summarise(monto\_venta = sum(as.numeric(montoVentaTotal), na.rm = TRUE))  
  
  
animalesVendidos <- ganaderia\_produccion %>%   
 group\_by(cons, grupo) %>%   
 # cantidad  
 summarise(animales\_venta = sum(as.numeric(s7c7dp02), na.rm = TRUE))  
  
  
ganaderia\_produccion <- left\_join(valorProduccionGanadera, animalesVendidos, by = c("cons" = "cons", "grupo" = "grupo"))  
  
head(ganaderia\_produccion,10)

## # A tibble: 10 x 9  
## # Groups: cons, edo, NOM\_ENT, mun, NOM\_MUN, grupo [10]  
## cons edo NOM\_ENT mun NOM\_MUN grupo NOMBRE\_GRUPO monto\_… anima…  
## <chr> <chr> <fctr> <chr> <fctr> <chr> <chr> <dbl> <dbl>  
## 1 0001 15 "M\xe9xico" 051 Lerma 1 reses 0 0  
## 2 0001 15 "M\xe9xico" 051 Lerma 2 equinos 0 0  
## 3 0001 15 "M\xe9xico" 051 Lerma 3 cerdos 0 0  
## 4 0001 15 "M\xe9xico" 051 Lerma 4 aves 0 0  
## 5 0001 15 "M\xe9xico" 051 Lerma 5 cyb 0 0  
## 6 0002 15 "M\xe9xico" 051 Lerma 1 reses 0 0  
## 7 0002 15 "M\xe9xico" 051 Lerma 2 equinos 0 0  
## 8 0002 15 "M\xe9xico" 051 Lerma 3 cerdos 0 0  
## 9 0002 15 "M\xe9xico" 051 Lerma 4 aves 0 0  
## 10 0002 15 "M\xe9xico" 051 Lerma 5 cyb 0 0

guardarTablaProcesada(ganaderia\_produccion)

En la tabla anterior se calculó el ingreso proveniente de la venta de animales, asi como la cantidad animales vendidos. Cuando el valor de ambas variables es cero, es porque los hogares no vendieron animales de ese grupo.

#### Mano de Obra (Labor)

La información de la mano de obra en el sector ganadero se obtiene de la tabla **Produccion\_ganade\_ta**, la cual distingue entre el trabajo realizado por miembros del hogar y el trabajo contratado.

En esta segunda base se incluyen preguntas de los siguientes cuadros:

* Cuadro 7E. Alimentación de los animales
* Cuadro 7F. Pastos e insumos
* Cuadro 7G. Mano de obra
* Cuadro 7H. Apoyo gubernamental
* Observaciones: Las preguntas de los cuadros 7E a 7H se aplican por grupo de animal

El siguiente código carga la tabla mencionada y enlista los campos contenidos en ella.

# Cargar base de datos en formato STATA a R y guardar base de datos en formato CSV.   
prod\_ganade\_ta <- procesarSTATA("Produccion\_ganade\_ta")  
# Desplegar el nombre de los campos  
names(prod\_ganade\_ta)

## [1] "edo" "mun" "loc"   
## [4] "cons" "s7c7ep01" "s7c7ep02"   
## [7] "s7c7ep03i" "s7c7ep04ai" "s7c7ep04bi"   
## [10] "s7c7ep04ci" "s7c7ep05i" "s7c7ep06i"   
## [13] "s7c7ep03ii" "s7c7ep04aii" "s7c7ep04bii"   
## [16] "s7c7ep04cii" "s7c7ep05ii" "s7c7ep06ii"   
## [19] "s7c7ep03iii" "s7c7ep04aiii" "s7c7ep04biii"   
## [22] "s7c7ep04ciii" "s7c7ep05iii" "s7c7ep06iii"   
## [25] "s7c7ep07" "s7c7ep08i" "s7c7ep09i"   
## [28] "s7c7ep10i" "s7c7ep11i" "s7c7ep12i\_lc"   
## [31] "s7c7ep12i\_lit" "s7c7ep13i\_mc" "s7c7ep13i\_mit"   
## [34] "s7c7ep14i\_edo" "s7c7ep08ii" "s7c7ep09ii"   
## [37] "s7c7ep10ii" "s7c7ep11ii" "s7c7ep12ii\_lc"   
## [40] "s7c7ep12ii\_lit" "s7c7ep13ii\_mc" "s7c7ep13ii\_mit"   
## [43] "s7c7ep14ii\_edo" "s7c7ep08iii" "s7c7ep09iii"   
## [46] "s7c7ep10iii" "s7c7ep11iii" "s7c7ep12iii\_lc"   
## [49] "s7c7ep12iii\_lit" "s7c7ep13iii\_mc" "s7c7ep13iii\_mit"  
## [52] "s7c7ep14iii\_edo" "s7c7fp01" "s7c7fp02"   
## [55] "s7c7fp03" "s7c7fp04" "s7c7fp05"   
## [58] "s7c7fp06" "s7c7fp07" "s7c7fp08"   
## [61] "s7c7gp01" "s7c7gp02" "s7c7gp03a"   
## [64] "s7c7gp03b" "s7c7gp03c" "s7c7gp03d"   
## [67] "s7c7gp03e" "s7c7gp04a" "s7c7gp04b"   
## [70] "s7c7gp04c" "s7c7gp04d" "s7c7gp04e"   
## [73] "s7c7gp04\_ca" "s7c7gp04\_cb" "s7c7gp04\_cc"   
## [76] "s7c7gp04\_cd" "s7c7gp04\_ce" "s7c7gp05a"   
## [79] "s7c7gp05b" "s7c7gp05c" "s7c7gp05d"   
## [82] "s7c7gp05e" "s7c7gp06a" "s7c7gp06b"   
## [85] "s7c7gp06c" "s7c7gp06d" "s7c7gp06e"   
## [88] "s7c7gp07" "s7c7gp08" "s7c7gp09"   
## [91] "s7c7gp09\_c" "s7c7gp10" "s7c7gp11"   
## [94] "s7c7gp12" "s7c7gp12\_c" "s7c7gp13"   
## [97] "s7c7hp01" "s7c7hp02" "s7c7hp03"   
## [100] "s7c7hp04" "s7c7hp05" "s7c7hp06"   
## [103] "s7c7hp07" "s7c7hp08" "s7c7hp09\_lc"   
## [106] "s7c7hp09\_lit" "s7c7hp10\_mc" "s7c7hp10\_mit"   
## [109] "s7c7hp11\_edo" "estrato" "ponderador"

Para objeto del MARCEG, la mano de obra se necesita calcular como las horas que el hogar le dedica a las actividades del sector ganadero.

El cuestionario **“Cuadro 7G. Mano de obra”** se encuentra dividido en *“MANO DE OBRA DEL HOGAR”* y *“MANO DE OBRA CONTRATADA”*, la primera considera miembros del hogar que se dedican a esta actividad y la segunda considera personas externas contratadas por el hogar para esta actividad.

Debido a que la tabla **Produccion\_ganade\_ta** reporta los resultados por miembro del hogar en formato de columnas, el proceso para limpiar la tabla de resultados es la siguiente:

1. Extraer una tabla con las respuestas de la sección de *“MANO DE OBRA DEL HOGAR”* y otra para *“MANO DE OBRA CONTRATADA”*.

Para cada una de estas nuevas tablas generadas:

1. Filtrar la pregunta de si hubo miembros del hogar que atendieron animales / personal contratado para atender animales.
2. Convertir de formato de columnas a formato de renglones la información de “horas y minutos”, “código horas y minutos”, “número de días a la semana” y “número de días al mes” que vienen reportadas por miembro del hogar A, B, C, D y E.
3. Calcular las horas de trabajo dedicadas por hogar y grupo de animal por día, semana, mes y año.

**a) Mano de obra de miembros del hogar**

A continuación se describe cómo obtener el número de horas totales al año de los miembros del hogar. La información se encuentra agregada por hogar y tipo de animal.

De la tabla **Produccion\_ganade\_ta** se construyen 4 tablas:

* horas y minutos
* código horas y minutos
* número de días a la semana
* número de meses en el año
* número de días al mes

Cada una de estas tablas se manipula para que cada miembro del hogar A, B, C, D y E que se encuentra en formato de columnas se convierta en un renglón. Adicionalmente, se construye una llave de miembros del hogar extrayendo la letra clave de lo que antes era el nombre del campo.

# Función para extraer el campo de miembro del hogar  
cleanKey <- function(key, start, end){  
 substring(key,start,end)  
}  
  
# horas y minutos  
ganaderia\_household\_labor\_time <- prod\_ganade\_ta %>%  
   
 filter(s7c7gp02 == 1) %>% # Trabajaron miembros del hogar en ganadería 1 sí / 2 no  
   
 select(  
 cons, # número de cuestionario  
 edo, # clave estado  
 mun, # clave municipio  
 loc, # clave localidad   
 s7c7gp01, # Grupo animal  
 s7c7gp04a, # Persona A: número horas o minutos al día le trabajó  
 s7c7gp04b, # Persona B: número horas o minutos al día le trabajó  
 s7c7gp04c, # Persona C: número horas o minutos al día le trabajó  
 s7c7gp04d, # Persona D: número horas o minutos al día le trabajó  
 s7c7gp04e # Persona E: número horas o minutos al día le trabajó  
 ) %>%  
   
 arrange(cons, s7c7gp01) %>%  
   
 gather(6:10, key = "householdMember", value = "num\_hours\_or\_min") %>%  
   
 mutate(member = cleanKey(householdMember,9,9)) %>%  
   
 mutate\_at(vars("cons","edo","mun","loc","s7c7gp01", "householdMember","member"), funs(as.character))

## Warning: attributes are not identical across measure variables;  
## they will be dropped

# codigo horas y minutos  
ganaderia\_household\_labor\_code\_time <- prod\_ganade\_ta %>%  
   
 filter(s7c7gp02 == 1) %>% # Trabajaron miembros del hogar en ganadería 1 sí / 2 no  
   
 select(  
 cons, # número de cuestionario   
 edo, # clave estado  
 mun, # clave municipio  
 loc, # clave localidad   
 s7c7gp01, # Grupo animal  
 s7c7gp04\_ca, # Persona A: código horas (1) o minutos (2)  
 s7c7gp04\_cb, # Persona B: código horas (1) o minutos (2)  
 s7c7gp04\_cc, # Persona C: código horas (1) o minutos (2)  
 s7c7gp04\_cd, # Persona D: código horas (1) o minutos (2)  
 s7c7gp04\_ce # Persona E: código horas (1) o minutos (2)  
 ) %>%  
   
 arrange(cons, s7c7gp01) %>%  
   
 gather(6:10, key = "householdMember", value = "code\_hours\_or\_min") %>%  
   
 mutate(member = cleanKey(householdMember,11,11)) %>%  
   
 mutate\_all(funs(as.character))

## Warning: attributes are not identical across measure variables;  
## they will be dropped

# número de días a la semana  
ganaderia\_household\_labor\_code\_days <- prod\_ganade\_ta %>%  
   
 filter(s7c7gp02 == 1) %>% # Trabajaron miembros del hogar en ganadería 1 sí / 2 no  
   
 select(  
 cons, # número de cuestionario  
 edo, # clave estado  
 mun, # clave municipio  
 loc, # clave localidad   
 s7c7gp01, # Grupo animal  
 s7c7gp05a, # Persona A: número de días a la semana  
 s7c7gp05b, # Persona B: número de días a la semana  
 s7c7gp05c, # Persona C: número de días a la semana  
 s7c7gp05d, # Persona D: número de días a la semana  
 s7c7gp05e # Persona E: número de días a la semana  
 ) %>%  
   
 arrange(cons, s7c7gp01) %>%  
   
 gather(6:10, key = "householdMember", value = "days") %>%  
   
 mutate(member = cleanKey(householdMember,9,9)) %>%  
   
 mutate\_at(vars("cons","edo","mun","loc","s7c7gp01", "householdMember", "member"),funs(as.character))

## Warning: attributes are not identical across measure variables;  
## they will be dropped

# número de días al mes  
ganaderia\_household\_labor\_code\_months <- prod\_ganade\_ta %>%  
   
 filter(s7c7gp02 == 1) %>% # Trabajaron miembros del hogar en ganadería 1 sí / 2 no  
   
 select(  
 cons, # número de cuestionario   
 edo, # clave estado  
 mun, # clave municipio  
 loc, # clave localidad   
 s7c7gp01, # Grupo animal  
 s7c7gp06a, # Persona A: número de meses  
 s7c7gp06b, # Persona B: número de meses  
 s7c7gp06c, # Persona C: número de meses  
 s7c7gp06d, # Persona D: número de meses  
 s7c7gp06e # Persona E: número de meses  
 ) %>%  
   
 arrange(cons, s7c7gp01) %>%  
   
 gather(6:10, key = "householdMember", value = "months") %>%  
   
 mutate(member = cleanKey(householdMember,9,9)) %>%  
   
 mutate\_at(vars("cons","edo","mun","loc", "s7c7gp01", "householdMember", "member"),funs(as.character))

## Warning: attributes are not identical across measure variables;  
## they will be dropped

Una vez que se cuentan con las 4 tablas estas se deben cruzar para formar una sola tabla llamada \*\* labor \*\* . El cruce se debe hacer en 4 etapas. Primero se hace un JOIN entre las tablas horas y minutos que se llamará labor; después en la nueva tabla labor se hace un JOIN con el código horas y minutos; después entre labor y días a la semana; y finalmente entre labor y el número de meses al año.

# Joins para formar una sola tabla de labor a partir de las cuatro tablas extraídas.  
ganaderia\_household\_labor <- left\_join(ganaderia\_household\_labor\_time, ganaderia\_household\_labor\_code\_time,   
 by = c("cons"= "cons", "s7c7gp01" = "s7c7gp01", "member" = "member"))  
  
ganaderia\_household\_labor <- left\_join(ganaderia\_household\_labor, ganaderia\_household\_labor\_code\_days,   
 by = c("cons"= "cons", "s7c7gp01" = "s7c7gp01", "member" = "member"))  
  
ganaderia\_household\_labor <- left\_join(ganaderia\_household\_labor, ganaderia\_household\_labor\_code\_months,   
 by = c("cons"= "cons", "s7c7gp01" = "s7c7gp01", "member" = "member"))  
  
ganaderia\_household\_labor <- left\_join(ganaderia\_household\_labor, cat\_gpo\_animal, by = c("s7c7gp01" ="COD\_GPO"))

Una vez que se realizados los JOINS se obtiene una tabla única con toda la información relevante. El siguiente paso es extraer solo los campos necesarios que se alimentan al modelo:

* Clave hogar
* Estado
* Municipio
* Localidad
* Clave del grupo del animal
* Nombre del grupo animal
* Miembro de la familia dedicado a la actividad

# horas or minutos

* clave horas o minutos
* días
* meses

# Extracción de campos relevantes  
ganaderia\_household\_labor <- select(ganaderia\_household\_labor,   
 cons,   
 edo.x, # clave estado  
 mun.x, # clave municipio  
 loc.x, # clave localidad   
 s7c7gp01,  
 NOMBRE\_GRUPO,  
 member,   
 num\_hours\_or\_min,   
 code\_hours\_or\_min,   
 days,   
 months) %>%  
 arrange(cons, s7c7gp01, member)

Una vez hechos los JOINS en la tabla labor, ahora es posible calcular para cada miembro del hogar el número de horas que le dedican a cada grupo animal.

# Función para construir un solo campo de tiempo en horas día  
calcularHora <- function(tiempo, codigoTiempo){  
 ifelse(codigoTiempo == 1, tiempo,   
 ifelse(codigoTiempo == 2, tiempo/60,   
 ifelse(is.na(tiempo), 0,   
 ifelse(is.nan(tiempo), 0, 0))))  
}  
  
# Cálculo de horas a la semana, mes y año, así como renombre de campos  
ganaderia\_household\_labor <- ganaderia\_household\_labor %>%   
 mutate(  
 horasDia = calcularHora(num\_hours\_or\_min, code\_hours\_or\_min),  
 horasSemana = horasDia \* days,   
 horasMes = horasSemana \* 4,  
 horasAno = horasMes \* months,  
 cons = sprintf("%04d", as.numeric(cons)),  
 edo.x = sprintf("%02d", as.numeric(edo.x)),  
 mun.x = sprintf("%03d", as.numeric(mun.x)),  
 loc.x = sprintf("%04d", as.numeric(loc.x))   
 ) %>%  
 rename(  
 cve\_edo = edo.x,  
 cve\_mun = mun.x,  
 cve\_loc = loc.x,  
 grupo\_animal = NOMBRE\_GRUPO,  
 miembro\_hogar = member  
 )  
# Joins de catalogos estado y municipio  
ganaderia\_household\_labor <- left\_join(ganaderia\_household\_labor,cat\_estado, by = c("cve\_edo" = "CVE\_ENT"))  
ganaderia\_household\_labor <- left\_join(ganaderia\_household\_labor, cat\_municipio, by = c("cve\_edo" = "CVE\_ENT", "cve\_mun" = "CVE\_MUN"))  
   
head(ganaderia\_household\_labor,10)

## # A tibble: 10 x 17  
## cons cve\_e… cve\_m… cve\_l… s7c7… grup… miemb… num\_h… code\_… days mont…  
## <chr> <chr> <chr> <chr> <chr> <chr> <chr> <dbl> <chr> <dbl> <dbl>  
## 1 0001 15 051 0060 4 aves a 1.00 1 7.00 12.0  
## 2 0001 15 051 0060 4 aves b 1.00 1 7.00 12.0  
## 3 0001 15 051 0060 4 aves c 1.00 1 7.00 12.0  
## 4 0001 15 051 0060 4 aves d NA <NA> NA NA   
## 5 0001 15 051 0060 4 aves e NA <NA> NA NA   
## 6 0010 15 051 0060 1 reses a 30.0 2 7.00 12.0  
## 7 0010 15 051 0060 1 reses b NA <NA> NA NA   
## 8 0010 15 051 0060 1 reses c NA <NA> NA NA   
## 9 0010 15 051 0060 1 reses d NA <NA> NA NA   
## 10 0010 15 051 0060 1 reses e NA <NA> NA NA   
## # ... with 6 more variables: horasDia <dbl>, horasSemana <dbl>, horasMes  
## # <dbl>, horasAno <dbl>, NOM\_ENT <fctr>, NOM\_MUN <fctr>

Finalmente, se agrega la información de la tabla por hogar y grupo animal.

# Agregación de la tabla trabajo para dar formato final.  
labor <- ganaderia\_household\_labor %>%  
   
 group\_by(cons, cve\_edo, NOM\_ENT, cve\_mun, NOM\_MUN, cve\_loc, grupo\_animal) %>%   
   
 summarise(horas\_anuales = sum(horasAno, na.rm = TRUE)) %>%  
   
 arrange(grupo\_animal)  
  
# Guardar tabla final  
guardarTablaProcesada(labor)  
# Mostrar primeros 10 renglones de la tabla final  
head(labor, 10)

## # A tibble: 10 x 8  
## # Groups: cons, cve\_edo, NOM\_ENT, cve\_mun, NOM\_MUN, cve\_loc [10]  
## cons cve\_edo NOM\_ENT cve\_mun NOM\_MUN cve\_loc grupo\_a… horas\_a…  
## <chr> <chr> <fctr> <chr> <fctr> <chr> <chr> <dbl>  
## 1 0001 15 "M\xe9xico" 051 Lerma 0060 aves 1008   
## 2 0009 15 "M\xe9xico" 051 Lerma 0060 aves 2352   
## 3 0014 15 "M\xe9xico" 051 Lerma 0060 aves 672   
## 4 0022 15 "M\xe9xico" 051 Lerma 0013 aves 616   
## 5 0036 15 "M\xe9xico" 051 Lerma 0013 aves 600   
## 6 0042 15 "M\xe9xico" 067 Otzolotepec 0018 aves 28.0  
## 7 0043 15 "M\xe9xico" 067 Otzolotepec 0018 aves 16.8  
## 8 0044 15 "M\xe9xico" 067 Otzolotepec 0018 aves 1344   
## 9 0047 15 "M\xe9xico" 067 Otzolotepec 0018 aves 336   
## 10 0048 15 "M\xe9xico" 067 Otzolotepec 0018 aves 840

# Número de filas  
nrow(labor)

## [1] 1784

El cuadro anterior muestra un extracto de la base de datos limpia de **a) Mano de obra de miembros del hogar** . La base de datos está organizada por hogar, grupo animal y horas al año dedicadas a la ganadería.

A continuación se muestra una tabla resumen del número de horas de mano de obra de los hogares dedicadas a la gandaería por grupo animal:

# Cuadro resumen  
labor %>% group\_by(grupo\_animal) %>% summarise(horas\_anuales = sum(horas\_anuales)) %>% arrange(desc(horas\_anuales))

## # A tibble: 6 x 2  
## grupo\_animal horas\_anuales  
## <chr> <dbl>  
## 1 aves 236168  
## 2 reses 194270  
## 3 equinos 183179  
## 4 cyb 168460  
## 5 cerdos 83896  
## 6 otro 10814

**b) Mano de obra contratada por el hogar**

Para obtener el monto total de horas contratadas por el hogar se realiza un proceso similar al de la sección anterior. Se definió una función para calcular el número total de horas a partir del número de trabajadores, el monto, el número de meses trabajados y la frecuencia del pago que reportó el jefe del hogar.

# Función que permite calcular el pago anual del total de hombres contratados  
calcularPagoTrabajoContratado <- function(trabajadores, pago, meses, frecuencia){  
   
 # caso semana  
 ifelse(frecuencia == 1, trabajadores \* pago \* 4 \* meses,  
 # caso mes   
 ifelse(frecuencia == 2, trabajadores \* pago \* meses,   
 # total  
 ifelse(frecuencia == 3, trabajadores \* pago,   
 # NA   
 ifelse(is.na(frecuencia), 0, 0))))  
 }  
  
# Pipiline que extrae hombres contrtados en el sector de la ganadería por el hogar y calculo del pago anual  
# Solo se aplicó a los hombres porque mujeres no había.  
ganaderia\_trabajo\_contratado <- prod\_ganade\_ta %>%  
   
 filter(s7c7gp07 == 1) %>%   
   
 select(cons, # número de cuestionario  
 edo, # clave estado  
 mun, # clave municipio  
 loc, # clave localidad  
 s7c7gp01, # Grupo animal  
   
 s7c7gp07, # Hubo mano de obra contratada por tipo de animal sí (1) no (2)  
   
 s7c7gp08, # num de hombres contratados // numeric  
 s7c7gp09, # dinero en pesos pagado por hombre // numeric  
 s7c7gp09\_c, # por semana (1), mes (2) o total (3)  
 s7c7gp10, # num de meses // numeric  
 s7c7gp11, # num de mujeres contratados  
 s7c7gp12, # dinero en pesos pagado por mjer  
 s7c7gp12\_c, # por semana (1), mes (2) o total (3)  
 s7c7gp13 # num de meses  
 ) %>%  
 mutate(  
 cons = sprintf("%04d", as.numeric(cons)),  
 edo = sprintf("%02d", as.numeric(edo)),  
 mun = sprintf("%03d", as.numeric(mun)),   
 loc = sprintf("%04d", as.numeric(loc))  
 ) %>%  
 mutate\_at(vars("s7c7gp01", "s7c7gp07", "s7c7gp09\_c", "s7c7gp12\_c"),funs(as.character)) %>%  
   
 mutate( pagoAnual = calcularPagoTrabajoContratado(  
 s7c7gp08, #trabajadores   
 s7c7gp09, #pago   
 s7c7gp10, #meses   
 s7c7gp09\_c)) #frecuencia  
  
#Guardar la tabla de factores de producción de la ganadería en la carpeta Bases\_Hogares\_Procesadas   
guardarTablaProcesada(ganaderia\_trabajo\_contratado)

Una vez calculado el pago anual de cada uno de los hogares por tipo de animal, se genera la tabla final. Se hacen los joins de los catálogos, se renombran los campos de interés, se agrupa por hogar, estado, municipio, localidad y grupo animal.

# Agregación de la tabla trabajo para dar formato final.  
hired\_labor <- ganaderia\_trabajo\_contratado %>%  
   
 group\_by(cons, edo, mun, loc, s7c7gp01) %>%   
   
 summarise(pago\_anual = sum(pagoAnual, na.rm = TRUE)) %>%  
   
 arrange(s7c7gp01)  
  
hired\_labor <- left\_join(hired\_labor,cat\_estado, by = c("edo" = "CVE\_ENT"))  
  
hired\_labor <- left\_join(hired\_labor,cat\_municipio, by = c("edo" = "CVE\_ENT", "mun" = "CVE\_MUN"))  
  
hired\_labor <- hired\_labor %>%  
 select(cons, edo, NOM\_ENT, mun, NOM\_MUN, loc, s7c7gp01, pago\_anual) %>%  
 rename(cve\_edo = edo,  
 cve\_mun = mun,  
 cve\_loc = loc,  
 cve\_animal = s7c7gp01)  
  
# Guardar tabla final  
guardarTablaProcesada(hired\_labor)  
# Mostrar primeros 10 renglones de la tabla final  
head(hired\_labor, 10)

## # A tibble: 10 x 8  
## # Groups: cons, cve\_edo, cve\_mun, cve\_loc [10]  
## cons cve\_edo NOM\_ENT cve\_mun NOM\_MUN cve\_l… cve\_… pago…  
## <chr> <chr> <fctr> <chr> <fctr> <chr> <chr> <dbl>  
## 1 0209 15 "M\xe9xico" 124 "San Jos… 0032 1 48000  
## 2 0563 12 Guerrero 001 "Acapulc… 0297 1 0  
## 3 1052 08 Chihuahua 007 Balleza 0001 1 3600  
## 4 1175 03 Baja California Sur 008 Los Cabos 1030 1 24000  
## 5 1177 03 Baja California Sur 008 Los Cabos 1030 1 24000  
## 6 1199 03 Baja California Sur 008 Los Cabos 0198 1 7200  
## 7 1222 04 Campeche 006 "Hopelch… 0057 1 3600  
## 8 1253 27 Tabasco 013 Nacajuca 0015 1 7200  
## 9 1255 27 Tabasco 013 Nacajuca 0015 1 20000  
## 10 1545 05 Coahuila de Zaragoza 030 Saltillo 0073 1 9600

A continuación se muestra un resumen del pago anual por clave animal. Es preciso mencionar que para esta pregunta la clave de los animales presenta un nivel de agregación superior.

hired\_labor %>% group\_by(cve\_animal) %>% summarise(pago\_anual = sum(pago\_anual)) %>% arrange(desc(pago\_anual))

## # A tibble: 3 x 2  
## cve\_animal pago\_anual  
## <chr> <dbl>  
## 1 1 266100  
## 2 5 26400  
## 3 2 12100

## Siguientes pasos

En este documento se ha presentado el procedimiento para procesar y limpiar la ENCHOR y obtener la informción relevante para el MARCEG. Como ejemplo se utilizó la ganadería, una de las actividades productivas de la economía rural. Utilizando la misma lógica se pueden obtener el resto de las actividades y las variables de interés.