

# Metodología

Bryan Castillo

17 de junio de 2021

## Metodología

El método consiste básicamente en obtener las bases de datos en formato *comma-separated values* (csv) a escala de personas de los censos 1992, 2002 y 2017 desde la página de internet del **Instituto Nacional de Estadísticas** (INE). Debido al gran esfuerzo de procesamiento que requiere procesar la gran cantidad de datos adquiridos, los archivos de datos censales para los tres periodos fueron cargados en el gestor de base de datos **PostgreSQL** a través de una librería, **Rpostgresql**, que sirve como intérprete entre los lenguajes de *R* y *SQL* y también como interfaz entre el *IDE RStudio* y el gestor de bases de datos **PostgreSQL**, de esta manera, además de cargar las bases de datos se pudieron elaborar las consultas pertinentes a los datos con ayuda de la librería **dbplyr**.

Con ayuda de las librerías mencionadas anteriormente se realizó la siguiente secuencia de pasos:

En primer lugar se configura y establece la conexión entre el lenguaje R y su interfaz gráfica RStudio con el gestor de base de datos PostgreSQL

```
fun_connect<-function(){dbConnect(RPostgres::Postgres(),
                                dbname = 'censos',
                                host = 'localhost',
                                port = 5432, # or any other port specified by your DBA
                                user = 'usuario',
                                password = 'contrasena',
                                options="-c search_path=censos")}}

conn <- fun_connect()
```

Luego se procede a crear tablas a partir de las características de los datos en formato *csv* es importante conocer el tipo de datos que se cargará, el número de filas, columnas y el tamaño de la celda que ocupara cada dígito o carácter, para esto se creo una función que hace un resumen del número máximo de caracteres de cada fila y columna de los datos a usar.

Devuelve el tamaño máximo de caracteres o dígitos que debe tener la variable:

```
test2002<-summarise_all(readRDS("C:/CEDEUS/Censos/Censo2002_Persona_Full.Rds"),funs(
  case_when(is.character(.)~max(nchar(.), na.rm=T),
            is.numeric(.) ~max(nchar(as.character(.)), na.rm=T))))

test1992<-summarise_all(readRDS("C:/CEDEUS/Censos/Censo1992_Persona_Full.Rds"),funs(
  case_when(is.character(.)~max(nchar(.), na.rm=T),
            is.numeric(.) ~max(nchar(as.character(.)), na.rm=T))))

test2017<-summarise_all(readRDS("C:/CEDEUS/Censos/Censo2017_Persona_Full.Rds"),funs(
  case_when(is.character(.)~max(nchar(.), na.rm=T),
            is.numeric(.) ~max(nchar(as.character(.)), na.rm=T))))
```

Devuelve la clase de la variable que se debe usar para crear la tabla:

```

abreColumnas<-function(x){
  tidyr::spread(dplyr::group_by(as.data.frame(summary.default(readRDS(x),
                                                    by=Var1))),key = Var2, value = Freq)}

censo1992<-abreColumnas("C:/CEDEUS/Censos/Censo1992_Persona_Full.Rds")
censo2002<-abreColumnas("C:/CEDEUS/Censos/Censo2002_Persona_Full.Rds")
censo2017<-abreColumnas("C:/CEDEUS/Censos/Censo2017_Persona_Full.Rds")

```

Creo una función que mezcla los dos *output* anteriores y da los parámetros para crear la tabla:

```

tablita<-function(x,y){
  a<-cbind(variables=row.names(as.data.frame(t(x))),as.data.frame(t(x)))

  rownames(a)<- 1:nrow(a)

  a<-left_join(y, a, by=c("Var1"="variables"))
  a<-a[,c(1,4,5)]
  colnames(a)<-c("variable", "clase", "largo")
  return(a)
}

censo_1992<-tablita(test1992, censo1992)
censo_2002<-tablita(test2002, censo2002)
censo_2017<-tablita(test2017, censo2017)

```

Creo esquema y me conecta a la base de datos:

```

#dbSendQuery(conn, "CREATE SCHEMA censos")

dbSendQuery(conn, "set search_path to censos;")

```

Creo una función que construye una consulta para crear las tablas con los datos generados a partir de la función tablita:

```

creaTabla<-function(x){
  y<-deparse(substitute(x))

  x$variable<-gsub("\\\\.", "", x$variable)
  x$variable<-stringr::str_replace_all(x$variable, "ñ", "n")
  x<-paste(c(paste("ID", "SERIAL PRIMARY KEY"),
              paste(
                paste(x$variable, x$clase),
                ("", x$largo, ""), "NOT NULL")), collapse=",")

  x<-paste0("CREATE TABLE ", y, " (", x, ")")
  return(x)}

dbSendQuery(conn, creaTabla(censo_1992))
dbSendQuery(conn, creaTabla(censo_2002))
dbSendQuery(conn, creaTabla(censo_2017))

```

Luego de crear las tablas creo una consulta que las llena:

```

dbSendQuery(conn, paste("copy censo_1992", creaCopy(censo_1992), " from PROGRAM '7z e
                        -so C:/CEDEUS/2021/abril1_ciudadesCosteras/input/censo1992.7z'
                        delimiter ',' csv header;"))

```

```
dbSendQuery(conn, paste("copy censo_2002", creaCopy(censo_2002), " from PROGRAM '7z e
                        -so C:/CEDEUS/2021/abril1_ciudadesCosteras/input/censo2002.7z'
                        delimiter ',' csv header;"))
dbSendQuery(conn, paste("copy censo_2012", creaCopy(censo_2012), " from PROGRAM '7z e
                        -so C:/CEDEUS/2021/abril1_ciudadesCosteras/input/censo2012.7z'
                        delimiter ',' csv header;"))
dbSendQuery(conn, paste("copy censo_2017", creaCopy(censo_2017), " from PROGRAM '7z e
                        -so C:/CEDEUS/2021/abril1_ciudadesCosteras/input/censo2017.7z'
                        delimiter ',' csv header;"))
```

Una vez cargados los datos en el gestor de base de datos **PostgreSQL** se empiezan a realizar consultas, en específico se quiere saber si las personas adultos mayores que habitan en la zona de estudio al momento de ser censadas residían en esa comuna o no, y si no preguntar de dónde venían.

En primer lugar se crearon subtablas que solo agrupan a los adultos mayores ( $\geq 60$  años) que residen en nuestra área de estudio

#### SET 1 DE COMUNAS:

*#1992*

```
dbSendQuery(conn, " CREATE TABLE comunas_censo1992 AS
SELECT *
FROM censo_1992
WHERE (comuna='05601' OR comuna='05602' OR comuna='05603' OR comuna='05604' OR
comuna='05604' OR comuna='05605' OR comuna='05606')AND edad>=60 ;")
```

*#2002*

```
dbSendQuery(conn, " CREATE TABLE comunas_censo2002 AS
SELECT *
FROM censo_2002
WHERE (comuna='05601' OR comuna='05602' OR comuna='05603' OR comuna='05604' OR
comuna='05604' OR comuna='05605' OR comuna='05606')AND p19c>=60 ;")
```

*#2017*

```
dbSendQuery(conn, " CREATE TABLE comunas_censo2017 AS
SELECT *
FROM censo_2017
WHERE (comuna='05601' OR comuna='05602' OR comuna='05603' OR comuna='05604' OR
comuna='05604' OR comuna='05605' OR comuna='05606')AND p09>=60 ;")
```

#### SET 2 DE COMUNAS:

*#1992*

```
dbSendQuery(conn, " CREATE TABLE com_censo1992_ii AS
SELECT *
FROM censo_1992
WHERE (comuna='05103' OR comuna='05107' OR comuna='05105' OR comuna='05405' OR
comuna='05403')AND edad>=60 ;")
```

*#2002*

```
dbSendQuery(conn, " CREATE TABLE com_censo2002_ii AS
SELECT *
FROM censo_2002
WHERE (comuna='05103' OR comuna='05107' OR comuna='05105' OR comuna='05405' OR
```

```

        comuna='05403')AND p19>=60 ;")

#2017
dbSendQuery(conn, " CREATE TABLE com_censo2017_ii AS
SELECT *
FROM censo_2017
WHERE (comuna='05103' OR comuna='05107' OR comuna='05105' OR comuna='05405' OR
comuna='05403')AND p09>=60 ;")

```

Luego a estas sub-tablas se les hace una consulta para saber cuantas personas vivían en otra comuna hace 5 años y en que comuna.

```
#### SET 1 DE COMUNAS:
```

```

test<-dbSendQuery(conn, "SELECT comuna, comuna_1987_origen3, COUNT(*)
FROM comunas_censo1992
GROUP BY comuna,comuna_1987_origen3;")

test<-dbFetch(test)

test$comuna_1987_origen3<-paste0("0",test$comuna_1987_origen3)

test<-split.data.frame(test,test$comuna)

# Cuantas personas fuera de esta comuna hace 5 años -----
fuera1992<-lapply(test,function(x){
  sum(x$count[x$comuna_1987_origen3!=unique(x$comuna)])/sum(x$count)*100}) %>%
  bind_rows(.) %>% pivot_longer(colnames(.))

# Cuantas personas vienen de cada comuna -----
comuna1992<-lapply(test, function(x){
  x[x$comuna_1987_origen3!=unique(x$comuna),] %>%
  group_by(comuna_1987_origen3) %>% summarise(n=sum(count))})%>%
  bind_rows(.,.id="Comuna")

# Censo 2002: conteos 24a por comuna -----

library(dplyr)

test<-dbSendQuery(conn, "SELECT comuna,p24a,p24b, COUNT(*)
FROM comunas_censo2002
GROUP BY comuna,p24a,p24b;")

test<-dbFetch(test)

#23A LUGAR DE RESIDENCIA HABITUAL
#23B CODIGO DEL PAIS O COMUNA DE RESIDENCIA HABITUAL
#24A LUGAR DE RESIDENCIA EN EL 97
#24B CODIGO DEL PAIS O COMUNA DE RESIDENCIA EN EL 97

test<-split.data.frame(test,test$comuna)

# Cuantas personas fuera de esta comuna hace 5 años -----
fuera2002<-lapply(test, function(x){

```

```

sum(x$count[x$p24a==2])/sum(x$count)*100}) %>%
bind_rows(.) %>% pivot_longer(colnames(.))

# Cuantas personas vienen de cada comuna -----
comuna2002<-lapply(test,function(x){
  x[x$p24a!=1,] %>% group_by( p24b) %>%
    summarise(n=sum(count))}) %>%
  bind_rows(.,.id="Comuna")

# Censo 2017: conteos p11 por comuna -----
test<-dbSendQuery(conn, "SELECT comuna,p11,p11comuna, COUNT(*)
FROM comunas_censo2017
GROUP BY comuna,p11,p11comuna;")

test<-dbFetch(test)

test<-split.data.frame(test,test$comuna)

# Cuantas personas fuera de esta comuna hace 5 años -----
fuera2017<-lapply(test, function(x){
  sum(x$count[x$p11!=2])/sum(x$count)*100}) %>%
  bind_rows(.) %>% pivot_longer(colnames(.))

# Cuantas personas vienen de cada comuna -----
comuna2017<-lapply(test,function(x){
  x[x$p11!=2,] %>% group_by(p11comuna) %>%
    summarise(n=sum(count))}) %>% bind_rows(.,.id="Comuna")

#### SET 2 DE COMUNAS:

test<-dbSendQuery(conn, "SELECT comuna, comuna_1987_origen3, COUNT(*)
FROM com_censo1992_ii
GROUP BY comuna,comuna_1987_origen3;")

test<-dbFetch(test)

test$comuna_1987_origen3<-paste0("0",test$comuna_1987_origen3)

test<-split.data.frame(test,test$comuna)

# Cuantas personas fuera de esta comuna hace 5 años -----
fuera1992_ii<-lapply(test,function(x){
  sum(x$count[x$comuna_1987_origen3!=unique(x$comuna)])/sum(x$count)*100}) %>%
  bind_rows(.) %>% pivot_longer(colnames(.))

# Cuantas personas vienen de cada comuna -----
comuna1992_ii<-lapply(test, function(x){
  x[x$comuna_1987_origen3!=unique(x$comuna),] %>%
    group_by(comuna_1987_origen3) %>%
    summarise(n=sum(count))}) %>% bind_rows(.,.id="Comuna")

# Censo 2002: conteos 24a por comuna -----

```

```

library(dbplyr)

test<-dbSendQuery(conn, "SELECT comuna,p24a,p24b, COUNT(*)
                        FROM com_censo2002_ii
                        GROUP BY comuna,p24a,p24b;")

test<-dbFetch(test)

#23A LUGAR DE RESIDENCIA HABITUAL
#23B CODIGO DEL PAIS O COMUNA DE RESIDENCIA HABITUAL
#24A LUGAR DE RESIDENCIA EN EL 97
#24B CODIGO DEL PAIS O COMUNA DE RESIDENCIA EN EL 97

test<-split.data.frame(test,test$comuna)

# Cuantas personas fuera de esta comuna hace 5 años -----
fuera2002_ii<-lapply(test, function(x){
  sum(x$count[x$p24a==2])/sum(x$count)*100}) %>%
  bind_rows(.) %>% pivot_longer(colnames(.))

# Cuantas personas vienen de cada comuna -----
comuna2002_ii<-lapply(test,function(x){
  x[x$p24a!=1,] %>% group_by( p24b) %>%
  summarise(n=sum(count))}) %>% bind_rows(.,.id="Comuna")

# Censo 2017: conteos p11 por comuna-----
test<-dbSendQuery(conn, "SELECT comuna,p11,p11comuna, COUNT(*)
                        FROM com_censo2017_ii
                        GROUP BY comuna,p11,p11comuna;")

test<-dbFetch(test)

test<-split.data.frame(test,test$comuna)

# Cuantas personas fuera de esta comuna hace 5 años -----
fuera2017_ii<-lapply(test, function(x){
  sum(x$count[x$p11!=2])/sum(x$count)*100}) %>%
  bind_rows(.) %>% pivot_longer(colnames(.))

# Cuantas personas vienen de cada comuna -----
comuna2017_ii<-lapply(test,function(x){
  x[x$p11!=2,] %>% group_by(p11comuna) %>%
  summarise(n=sum(count))}) %>% bind_rows(.,.id="Comuna")

```

Una vez contenida la información en tablas se crea una función que grafica en heatmaps las comunas de origen y destino:

```

#### SET 1 DE COMUNAS:

graficOrigen<-function(x,y){
  ggplot(x, aes(y=Comuna_actual,x=Comuna_5años,
    fill=as.integer(Cantidad)))+
  geom_tile()+

```

```

theme(axis.text.y = element_text(size=8),axis.text.x = element_text(size=6,angle=90,
hjust=1,vjust=0.5))+
scale_fill_distiller(palette = "YlOrRd", direction=1)+
labs(x="Comunas de Origen", y="Comunas costeras",
title = paste0("Comuna donde residían las personas que actualmente viven
comunas costeras en el año ", y) , fill= "Cantidad de personas")+
scale_y_discrete(labels=c("San Antonio", "Algarrobo", "Cartagena","El Quisco",
"El Tabo", "Santo Domingo"))
}

g1992<-graficOrigen(comuna1992_ii,"1987") %>% plotly::ggplotly(.,originalData=F)
g2002<-graficOrigen(comuna2002_ii, "1997")%>% plotly::ggplotly(.)
g2017<-graficOrigen(comuna2017_ii, "2012")%>% plotly::ggplotly(.)

#### SET 2 DE COMUNAS:

graficOrigen_ii<-function(x,y){
  ggplot(x, aes(y=Comuna_actual,x=Comuna_5años, fill=as.integer(Cantidad)))+
  geom_tile()+
  theme(axis.text.y = element_text(size=8),axis.text.x = element_text(size=6,angle=90,
hjust=1,vjust=0.5))+
  scale_fill_distiller(palette = "YlOrRd", direction=1)+
  labs(x="Comunas de Origen", y="Comunas costeras",
title = paste0("Comuna donde residían las personas que actualmente viven
comunas costeras en el año ", y) , fill= "Cantidad de personas")+
  scale_y_discrete(labels=c("Concón", "Quintero", "Puchuncaví", "Papudo","Zapallar"))
}

g1992_ii<-graficOrigen_ii(comuna1992_ii,"1987") %>% plotly::ggplotly(.,originalData=F)
g2002_ii<-graficOrigen_ii(comuna2002_ii, "1997")%>% plotly::ggplotly(.)
g2017_ii<-graficOrigen_ii(comuna2017_ii, "2012")%>% plotly::ggplotly(.)

```

Se procedió a crear tablas dónde se pone el número y porcentaje de personas  $\geq 60$  años que vivían en otra comuna al momento de ser encuestadas y a crear “*heatmaps*” y “*alluvial diagrams*” para sintetizar la información que nos dice de que comuna vienen las personas que hace 5 años vivían en otra comuna.