Tarea 2

Análisis de Datos I

Enzo Loiza B. - 23 de Abril 2024

Introducción

Consideramos un extracto de las respuestas de la Encuesta de Caracterización Socioeconómica CASEN de 2017 adjunta en el archivo `casen.dta` con las siguientes variables:

Variables a utilizar de la Encuesta CASEN 2017

Variables	Descripción
`region`	Número de la región
`comuna`	Código de la Comuna
`tot_hog`	Total de hogares en la vivienda
`tot_per`	Total de personas en el hogar
`tot_nuc`	Total de núcleos en el hogar
`sexo`	1 (Hombre), 2 (Mujer)
`edad`	Edad en años
`ecivil`	Estado civil
`e9te`	Tipo de enseñanza
`s5`	Edad al tener el primer hijo
`s12`	Sistema de salud al que pertenece
`qaut`	Quintil autónomo nacional

Pasos previos

Abrimos las librerías necesarias para el análisis.

```
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(tidyverse)
```

```
## ── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
## ✓ forcats   1.0.0     ✓ readr     2.1.5
## ✓ ggplot2   3.5.0     ✓ stringr   1.5.1
## ✓ lubridate 1.9.3     ✓ tibble    3.2.1
## ✓ purrr     1.0.2     ✓ tidyr     1.3.1
```

```
## ── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
## ✖ dplyr::filter() masks stats::filter()
## ✖ dplyr::lag()    masks stats::lag()
## ℹ Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to be
come errors
```

```
library(ggplot2)
library(ggthemes)
```

Luego leemos los datos de la encuesta y hacemos una primera vista de ellos.

```
db <- read.csv('casen.csv')
head(db)
```

	region <int>	comuna <int>	tot_hog <int>	tot_per <int>	tot_nuc <int>	sexo <int>	edad <int>	ecivil <int>	e9te <int>
1	1	1101	1	1	1	2	56	8	*NA*
2	1	1101	1	1	1	2	21	8	*NA*
3	1	1101	1	2	1	1	24	2	*NA*
4	1	1101	1	2	1	1	28	2	*NA*
5	1	1101	1	3	1	1	26	1	*NA*
6	1	1101	1	3	1	2	26	1	*NA*

6 rows | 1-10 of 13 columns

Pregunta 1

a. Filtramos usando el comando `filter` de la base de datos, sólo las personas de la Región de Magallanes y la Antártica Chilena. De acuerdo al Libro de Códigos de la CASEN, esta región corresponde al número 12.

```
magallanes <- filter(db, region == 12)
head(magallanes)
```

	region <int>	comuna <int>	tot_hog <int>	tot_per <int>	tot_nuc <int>	sexo <int>	edad <int>	ecivil <int>	e9te <int>
1	12	12101	1	2	1	1	62	1	*NA*
2	12	12101	1	2	1	2	61	1	*NA*
3	12	12101	1	3	1	1	64	1	*NA*
4	12	12101	1	3	1	2	28	8	*NA*
5	12	12101	1	3	1	2	55	1	998
6	12	12101	1	2	1	1	49	1	*NA*

6 rows | 1-10 of 13 columns

Luego, usamos la librería `dplyr` para, en cadena, hacer el reporte de mínimo, máximo, media y desviación estándar de la variable `tot_hog` (número de hogares por vivienda) respecto de `qaut` (quintil autónomo nacional).

```
resumen <- magallanes %>%
  group_by(qaut) %>%
  summarise(
    Minimo = min(tot_hog),
    Maximo = max(tot_hog),
    Media = mean(tot_hog),
    Desviacion_Estandar = sd(tot_hog)
  )

resumen
```

qaut <int>	Minimo <int>	Maximo <int>	Media <dbl>	Desviacion_Estandar <dbl>
1	1	3	1.012212	0.1216401
2	1	4	1.033934	0.2231651
3	1	4	1.019074	0.2013349
4	1	4	1.041194	0.2723384
5	1	7	1.041414	0.3405342
NA	1	1	1.000000	0.0000000

6 rows

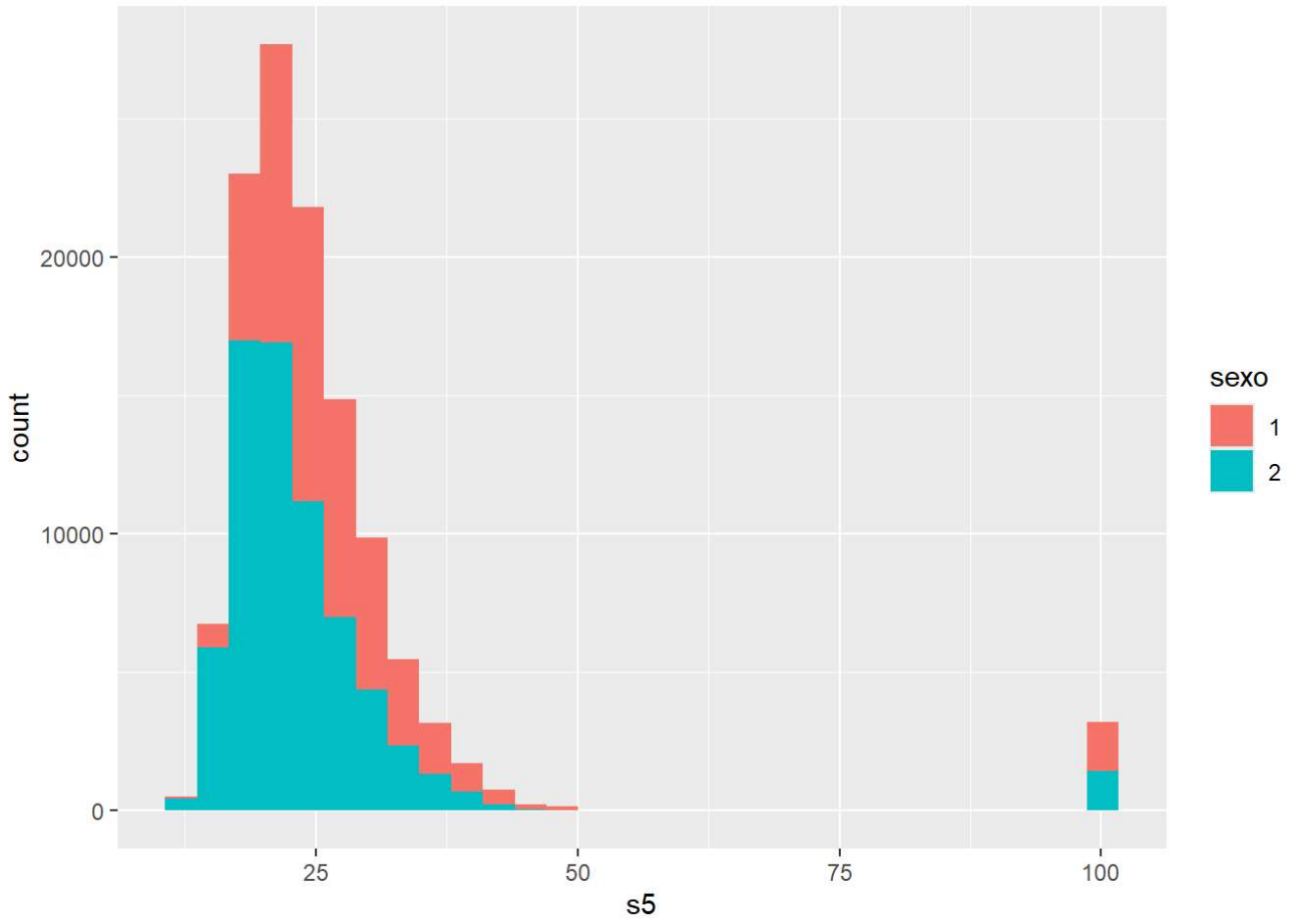
b. Sin considerar los filtros anteriores, reportamos la edad media de las personas al tener su primer hijo. Acá es importante restar de la base de datos a quienes aparecen como `NA's` , pues corresponden en su mayoría a personas que no (o aún no) tienen hijos.

Primero realizamos una vista de cómo se comportan los datos:

```
db$sexo <- as.factor(db$sexo)
ggplot(db, aes(x = s5, fill = sexo)) +
  geom_histogram()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```

```
## Warning: Removed 97067 rows containing non-finite outside the scale range
## (`stat_bin()`).
```



Esto nos indica que además de los 97067 casos sin edad reportada, existen también algunos casos donde la edad se dispara. Generalmente los 99 también son instancias sin datos, por lo que es necesario verificar si se trata de ellos.

```
db %>% filter(s5 > 75) %>% # filtramos por una edad prudente de 75
  summarise(
    min = min(s5),
    max = max(s5),
    mean = mean(s5),
    n = length(s5)
  )
```

min <int>	max <int>	mean <dbl>	n <int>
99	99	99	3203

1 row

Como se comprobó, que es así, ahora realizamos los filtros que corresponden para obtener la media de edad:

```
primer_hijo_1 <- db %>%
  filter(s5 != 99) %>%
  summarise(
    Media_1 = mean(s5, na.rm = TRUE)
)

primer_hijo_1
```

Media_1 <dbl>
23.59679

1 row

Ahora filtramos seleccionando todas las mujeres casadas (de acuerdo al Libro de Códigos, corresponde al 2) que tuvieron enseñanza básica. En el mismo documento corresponde a las personas que:

- han tenido más estudios que Educación Básica, es decir superan la educación parvularia y en el filtro se entiende como `e9te != 10` ,
- se pueden clasificar, es decir `e9te != 77` ,
- y no son blancos.

Luego,

```
primer_hijo_2 <- db %>% filter(ecivil == 2) %>%
  filter(sexo == 2, na.rm = TRUE) %>%
  filter(e9te != 10) %>%
  filter(e9te != 77) %>%
  filter(s5 != 99) %>%
  summarise(
    Media_2 = mean(s5, na.rm = TRUE)
  )

primer_hijo_2
```

Media_2 <dbl>
20.34834

1 row

Que es ligeramente menor que la media de edad a nivel global. Esto al parecer tiene que ver con que un gran porcentaje de personas tiene los datos de colegiatura en blanco. Esto se explica con la siguiente tabla:

```
db %>% filter(ecivil == 2 ) %>%
  filter(s5 != 99) %>%
  group_by(e9te) %>%
  summarise(mean = mean(s5, na.rm = TRUE),
            n = length(s5))
```

e9te <int>	mean <dbl>	n <int>
110	18.93750	16
310	18.40860	93
410	17.29412	17
998	21.89205	528
NA	23.17058	21439

5 rows

Pregunta 2

a. Se pregunta por la probabilidad de que una persona tenga sistema de previsión FONASA de cualquier grupo.

```
personas_fonasa <- filter(db, s12 %in% 1:5)
total_personas <- nrow(db)
total_fonasa <- nrow(personas_fonasa)

probabilidad_fonasa <- total_fonasa / total_personas
print(paste("Probabilidad de tener FONASA:", probabilidad_fonasa))
```

```
## [1] "Probabilidad de tener FONASA: 0.799957493797329"
```

b. Estimamos la cantidad de núcleos en el hogar que reportan las personas que se encuentran en el 10% inferior de la muestra del siguiente modo.

Primero ordenamos la base de datos en orden ascendente respecto de la variable `tot_nuc` , y luego identificamos el 10% inferior de la muestra de la base,

```
db_ascendente <- arrange(db, tot_nuc)

inferior <- round(0.1*nrow(db_ascendente))
```

Luego, seleccionamos el 10% inferior y obtenemos los datos de esta muestra

```
db_inf <- db_ascendente[1:inferior, ]
db_inf %>% summarise(
  min = min(tot_nuc),
  max = max(tot_nuc),
  mean = mean(tot_nuc)
)
```

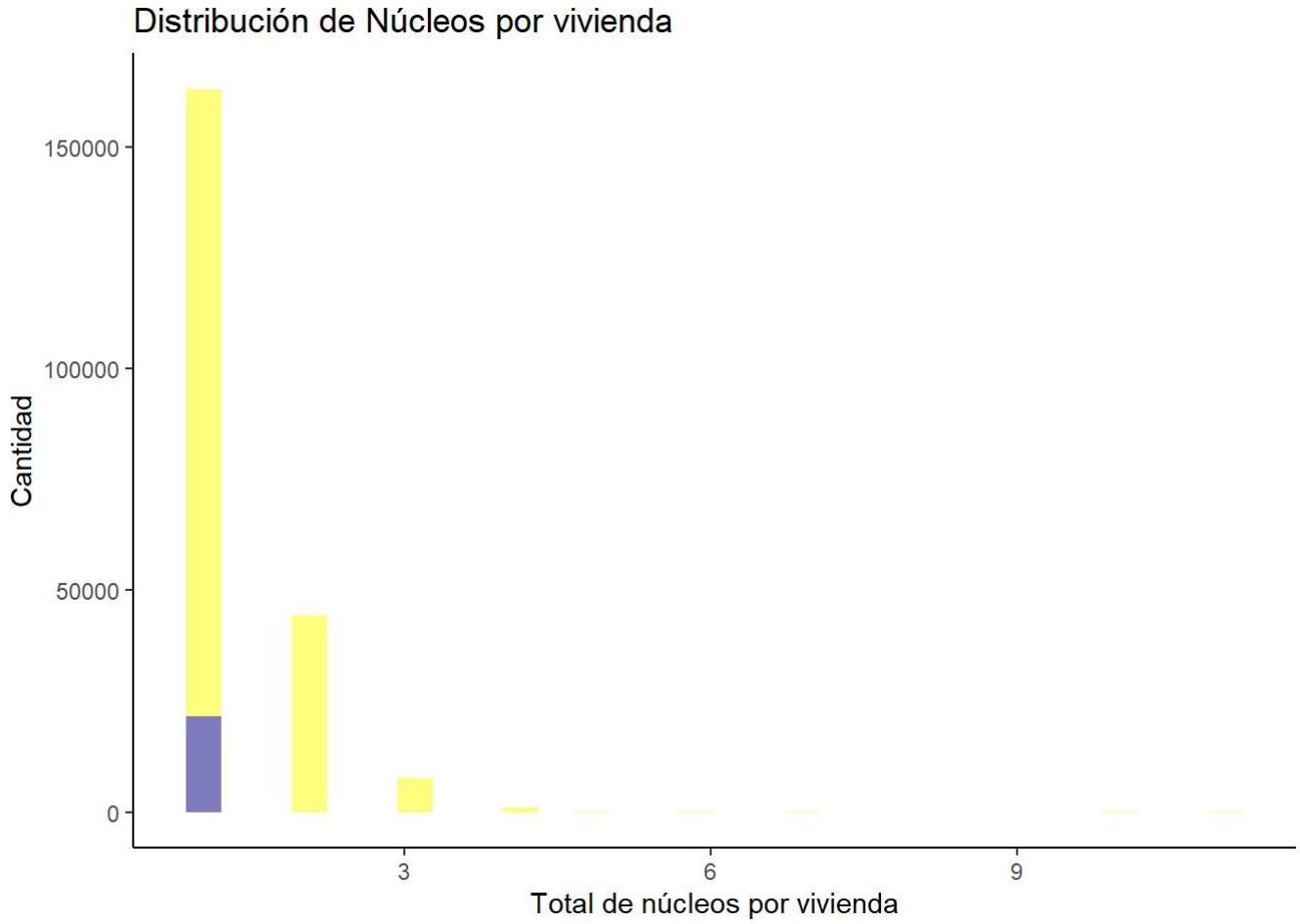
min <int>	max <int>	mean <dbl>
1	1	1

1 row

Es decir, el 10% inferior de la muestra tiene un solo núcleo por vivienda. Esto también podemos graficarlo del siguiente modo.

```
ggplot(db_ascendente,
       aes(x = tot_nuc)) +
  geom_histogram(fill = 'yellow',
                 alpha = .5) +
  geom_histogram(data = db_inf,
                 aes(x = tot_nuc),
                 fill = 'blue',
                 alpha = .5) +
  labs(title = "Distribución de Núcleos por vivienda",
       x = "Total de núcleos por vivienda",
       y = "Cantidad") +
  theme_classic()
```

```
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
## `stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
```



Disponible en html y en GitHub (https://github.com/zoendeloi/MPP-ADI).