SOC4001 Procesamiento avanzado de bases de datos en R

Tarea 2, respuestas

Ponderación: 12% de la nota final del curso

Formato: Desarrollar esta tarea en un RScript, agregando comentarios cuando sea necesario.

1) Carga la base de datos "Chile" del paquete `carData` y crea un objeto que los contenga los datos. Llama tal objeto "datos_chile". Carga la librería `tidyverse` y ejecuta la siguientes operaciones usando las herramientas contenidas de `tidyverse`:

```
library("carData")
library("tidyverse")
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v ggplot2 3.3.6     v purrr   0.3.4
## v tibble  3.1.8     v dplyr   1.0.9
## v tidyr   1.2.0     v stringr 1.4.0
## v readr   2.1.2     v forcats 0.5.1
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
data("Chile")
datos_chile <- Chile
rm(Chile) # remueve "flotante"

datos_chile %>% glimpse()
```

```
## Rows: 2,700
## Columns: 8
## $ region     <fct> N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N,~
## $ population <int> 175000, 175000, 175000, 175000, 175000, 175000, 175000, 175~
## $ sex        <fct> M, M, F, F, F, F, M, F, F, M, M, M, F, F, M, M, F, M, M, F,~
## $ age        <int> 65, 29, 38, 49, 23, 28, 26, 24, 41, 41, 64, 19, 27, 46, 36,~
## $ education  <fct> P, PS, P, P, S, P, PS, S, P, P, P, S, PS, S, PS, S, PS, S, ~
## $ income     <int> 35000, 7500, 15000, 35000, 35000, 7500, 35000, 15000, 15000~
## $ statusquo  <dbl> 1.00820, -1.29617, 1.23072, -1.03163, -1.10496, -1.04685, -~
## $ vote       <fct> Y, N, Y, N, N, N, N, N, U, N, Y, U, Y, Y, NA, A, N, U, Y, U~
```

2) Añade a "datos_chile" un variable llamada "year" con valor 1988 en todas las filas

```
datos_chile <- datos_chile %>% mutate(year = 1988)
```

3) Calcula el año de nacimiento de cada individuo. Añade a "datos_chile" un variable llamada "birthyear" que contenga esta información

```
datos_chile <- datos_chile %>% mutate(birthyear = year - age)
```

4) Usando la función `if_else()` añade a "datos_chile" un variable llamada "vote_no" que tome valor 1 si la persona declara que votará por el No y valor 0 en cualquier otra caso.

```
datos_chile <-  datos_chile %>% mutate(vote_no = if_else(vote=="N",1,0))
```

5) Usando la función `case_when()` añade a "datos_chile" un variable llamada "cohort73" que tome valor 1 si la persona tenía 18 año o más el año del golpe de estado (1973) y valor 0 si tenía menos de 18. Trata las observaciones que no cumplan ninguna de estas condiciones como valores perdidos.

```
datos_chile <-  datos_chile %>% mutate(cohort73 = case_when(birthyear <= (1973 - 18) ~ 1,
                              birthyear > (1973 - 18) ~ 0)
                            )
```

6) Usando la función `group_by()` añade a "datos_chile" un variable llamada "no_by_groups" que contenga el promedio de la variable "vote_no" por región, nivel educacional y cohorte (cohort73).

```
datos_chile <-  datos_chile %>% group_by(region,education,cohort73) %>%
                mutate(no_by_groups = mean(vote_no, na.rm = T))
```

```
datos_chile %>% select(no_by_groups) %>% glimpse()
```

```
## Adding missing grouping variables: `region`, `education`, `cohort73`
```

```
## Rows: 2,700
## Columns: 4
## Groups: region, education, cohort73 [35]
## $ region       <fct> N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, N, ~
## $ education    <fct> P, PS, P, P, S, P, PS, S, P, P, P, S, PS, S, PS, S, PS, S~
## $ cohort73     <dbl> 1, 0, 1, 1, 0, 0, 0, 0, 1, 1, 1, 0, 0, 1, 1, 0, 0, 0, 1, ~
## $ no_by_groups <dbl> 0.2020202, 0.5312500, 0.2020202, 0.2020202, 0.3939394, 0.~
```

7) Usando la funciones `summarise()` y `group_by()` calcula el promedio de la variable "vote_no" por región, nivel educacional y cohorte (cohort73) y almacénalo en una variable llamada "datos_chile". Asigna el resultado a un nuevo objeto llamado `resultados`.

```
resultados <-  datos_chile %>% group_by(region,education,cohort73) %>%
               summarise(no_by_groups = mean(vote_no, na.rm = T))
```

```
## `summarise()` has grouped output by 'region', 'education'. You can override
## using the `.groups` argument.
```

```
print(resultados, n=40)
```

```
## # A tibble: 35 x 4
## # Groups:   region, education [19]
##    region education cohort73 no_by_groups
```

```
##    <fct>  <fct>        <dbl>        <dbl>
##  1 C      P                0        0.333
##  2 C      P                1        0.270
##  3 C      PS               0        0.6
##  4 C      PS               1        0.649
##  5 C      S                0        0.388
##  6 C      S                1        0.356
##  7 C      <NA>             1        0
##  8 M      P                0        0.25
##  9 M      P                1        0.152
## 10 M      PS               0        0.333
## 11 M      PS               1        0
## 12 M      S                0        0.25
## 13 M      S                1        0.4
## 14 N      P                0        0.222
## 15 N      P                1        0.202
## 16 N      PS               0        0.531
## 17 N      PS               1        0.419
## 18 N      S                0        0.394
## 19 N      S                1        0.375
## 20 N      <NA>             1        0
## 21 S      P                0        0.271
## 22 S      P                1        0.2
## 23 S      PS               0        0.5
## 24 S      PS               1        0.48
## 25 S      S                0        0.391
## 26 S      S                1        0.317
## 27 S      <NA>             1        1
## 28 SA     P                0        0.378
## 29 SA     P                1        0.296
## 30 SA     P               NA        0
## 31 SA     PS               0        0.505
## 32 SA     PS               1        0.495
## 33 SA     S                0        0.468
## 34 SA     S                1        0.325
## 35 SA     <NA>             1        0.167
```

8) Usando la funciones `summarise()`, `across()`) `group_by()` calcula el promedio y la deviación estándar de las variables "vote_no" e "income" por región, nivel educacional y cohorte (cohort73). Asigna el resultado a un nuevo objeto llamado `resultados`.

```
resultados <-  datos_chile %>% group_by(region,education,cohort73) %>%
                summarise(across(c("vote_no", "income"), list(media = ~mean(.x, na.rm = TRUE), sd= ~sd(
```

```
## `summarise()` has grouped output by 'region', 'education'. You can override
## using the `.groups` argument.
```

```
print(resultados, n=40)
```

```
## # A tibble: 35 x 7
## # Groups:   region, education [19]
##    region education cohort73 vote_no_media vote_no_sd income_media income_sd
```

```
##    <fct>  <fct>        <dbl>        <dbl>      <dbl>       <dbl>     <dbl>
##  1 C      P                0        0.333      0.475      14683.    11786.
##  2 C      P                1        0.270      0.445      16740.    20026.
##  3 C      PS               0        0.6        0.496      62692.    46747.
##  4 C      PS               1        0.649      0.484      64812.    53471.
##  5 C      S                0        0.388      0.489      30581.    27994.
##  6 C      S                1        0.356      0.481      44231.    47286.
##  7 C      <NA>             1        0          0          41250     47730.
##  8 M      P                0        0.25       0.452      17292.    13877.
##  9 M      P                1        0.152      0.364      21776.    22284.
## 10 M      PS               0        0.333      0.577      48333.    23094.
## 11 M      PS               1        0          0          71667.    55076.
## 12 M      S                0        0.25       0.444      30000     23979.
## 13 M      S                1        0.4        0.516      27333.    17384.
## 14 N      P                0        0.222      0.422      17014.    13280.
## 15 N      P                1        0.202      0.404      20248.    17326.
## 16 N      PS               0        0.531      0.507      43548.    30026.
## 17 N      PS               1        0.419      0.502      61953.    49108.
## 18 N      S                0        0.394      0.492      30469.    23854.
## 19 N      S                1        0.375      0.489      34796.    29531.
## 20 N      <NA>             1        0         NA          15000        NA
## 21 S      P                0        0.271      0.447      13737.    14266.
## 22 S      P                1        0.2        0.401      16910.    20691.
## 23 S      PS               0        0.5        0.505      49600     52346.
## 24 S      PS               1        0.48       0.505      62260.    51324.
## 25 S      S                0        0.391      0.490      26786.    29576.
## 26 S      S                1        0.317      0.468      34205.    30228.
## 27 S      <NA>             1        1         NA           2500        NA
## 28 SA     P                0        0.378      0.492      16447.    21838.
## 29 SA     P                1        0.296      0.457      20133.    15187.
## 30 SA     P               NA        0         NA          15000        NA
## 31 SA     PS               0        0.505      0.503      73672.    63285.
## 32 SA     PS               1        0.495      0.503      89747.    67425.
## 33 SA     S                0        0.468      0.500      35631.    36453.
## 34 SA     S                1        0.325      0.470      45803.    44526.
## 35 SA     <NA>             1        0.167      0.408      10500      4108.
```