OPSO79-1-UCSH2021

Tarea 3, case_when() y prácticas para una buena codificación.

08/10/2021

Revisión Tarea 3

Utilizando la Encuesta Suplementaria de Ingresos (ESI)

Tarea 3

Cargar la data

```
library(haven)
library(dplyr)
```

```
esi <- read_sav("data/esi-2019---personas_s.sav")
```

1. ¿Cuántas personas en Chile tienen 65 o más años?

Rbase

```
pregunta1 <- esi[esi$edad>=65,]
nrow(pregunta1)
```

```
## [1] 15092
```

```
esi %>% filter(edad>=65) %>% nrow() ## dplyr
```

Tarea 3

2. ¿Cuál es el promedio de ingresos de hombres y mujeres? (use la variable `ing_mon_sb` y la función `group_by()`). Previamente remueva los valores cero de la variable de ingresos.

Sin sacar los ceros

```
esi %>% group_by(sexo) %>% summarise(mean(ing_mon_sb))
```

```
## # A tibble: 2 x 2
##         sexo `mean(ing_mon_sb)`
##    <dbl+lbl>              <dbl>
## 1 1 [Hombre]            377027.
## 2 2 [Mujer]             227941.
```

Sacando los ceros

```
esi %>%
  filter(ing_mon_sb>0) %>%
  group_by(sexo) %>%
  summarise(mean(ing_mon_sb))
```

Tarea 3

3. Reste el promedio de ingresos entre hombres y mujeres. ¿Cuál es la brecha de ingresos entre ambos sexos?

Guardar como objeto tabla anterior

```
pregunta2 <- esi %>%
  filter(ing_mon_sb>0) %>%
  group_by(sexo) %>%
  summarise(mean(ing_mon_sb))
```

```
pregunta2[2,2] - pregunta2[1,2]
```

```
##   mean(ing_mon_sb)
## 1        -204102.8
```

Tarea 3

4. Crea 2 tablas con el promedio de ingresos (cualquier variable de ingresos) según región y sector económico

```
esi %>%
  group_by(region) %>%
  summarise(mean(ing_mon_sb))
```

```
## # A tibble: 16 x 2
##                                                    region `mean(ing_mon_sb)`
##                                                  <dbl+lbl>              <dbl
##  1  1 [Región de Tarapacá]                                              296522
##  2  2 [Región de Antofagasta]                                           417337
##  3  3 [Región de Atacama]                                               284777
##  4  4 [Región de Coquimbo]                                              249269
##  5  5 [Región de Valparaíso]                                            306519
##  6  6 [Región del Libertador Gral. Bernardo O'Higgins]                  252775
##  7  7 [Región del Maule]                                                242156
##  8  8 [Región del Biobío]                                               231571
##  9  9 [Región de La Araucanía]                                          239898
## 10 10 [Región de Los Lagos]                                             256319
## 11 11 [Región de Aysén del Gral. Carlos Ibáñez del Campo]               363213
```

Tarea 3

4. Crea 2 tablas con el promedio de ingresos (cualquier variable de ingresos) según región y sector económico

```
esi %>%
  group_by( b13_rev4cl_caenes) %>%
  summarise(mean(ing_mon_sb))
```

```
## # A tibble: 22 x 2
##                                                b13_rev4cl_caenes `mean(ing_mo
##                                                        <dbl+lbl>
##  1  1 [Agricultura, ganadería, silvicultura y pesca]                        29
##  2  2 [Explotación de minas y canteras]                                     83
##  3  3 [Industrias manufactureras]                                           67
##  4  4 [Suministro de electricidad, gas, vapor y aire acondici~              64
##  5  5 [Suministro de agua; evacuación de aguas residuales, ge~              47
##  6  6 [Construcción]                                                        60
##  7  7 [Comercio al por mayor y al por menor; reparación de ve~              50
##  8  8 [Transporte y almacenamiento]                                         56
##  9  9 [Actividades de alojamiento y de servicio de comidas]                 45
## 10 10 [Información y comunicaciones]                                        85
## # ... with 12 more rows
```

Tarea 3

5. Identifica para cada combinación de región y de sector económico el total de casos en el microdato (solo 1 tabla)

```
esi %>%
  group_by(region,b13_rev4cl_caenes) %>%
  summarise(n())
```

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

```
## # A tibble: 244 x 3
## # Groups:   region [16]
##                    region                                  b13_rev4cl_caenes
##                  <dbl+lbl>                                       <dbl+lbl>
##  1 1 [Región de Tarapacá]  2 [Explotación de minas y canteras]
##  2 1 [Región de Tarapacá]  3 [Industrias manufactureras]
##  3 1 [Región de Tarapacá]  5 [Suministro de agua; evacuación de aguas res~
##  4 1 [Región de Tarapacá]  6 [Construcción]
##  5 1 [Región de Tarapacá]  7 [Comercio al por mayor y al por menor; repar~
##  6 1 [Región de Tarapacá]  8 [Transporte y almacenamiento]
##  7 1 [Región de Tarapacá]  9 [Actividades de alojamiento y de servicio de~
##  8 1 [Región de Tarapacá] 12 [Actividades inmobiliarias]
```

Tarea 3

6. DESAFÍO if_else(): ¿Cuántas personas tienen entre 0 y 17 años, entre 18 y 29 años, entre 30 y 64 años, y 65 o más años? (cuatro grupos)

```
esi <- esi %>%
  mutate(edad_recod=if_else(edad<18,"menos de 18",""),
         edad_recod=if_else(edad>17 & edad < 30,"entre 18 y 29",edad_rec
         edad_recod=if_else(edad>29 & edad < 65,"entre 30 y 64",edad_rec
         edad_recod=if_else(edad>64,"65 o más",edad_recod))
```

```
table(esi$edad_recod)
```

```
##
##      65 o más entre 18 y 29 entre 30 y 64   menos de 18
##         15092         15938         42967         22243
```

```
esi %>% group_by(edad_recod) %>% summarise(n())
```

Tarea 3

7. Crea una nueva data donde cada unidad (fila) sean los hogares y que solo tenga 3 variables: identificador del hogar, decil de los ingresos del hogar y región en la que se ubica el hogar (pista: use `group_by()` y `slice()`).

```
# sjlabelled::get_label(esi) Para identificar la variable
hogares <- esi %>%
  select(id_identificacion,decilh_sb,region) %>%
  group_by(id_identificacion) %>%
  slice(1)
```

Tarea 3

8. ¿Cuántos hogares son?, ¿Cómo se distribuyen regionalmente estos hogares?

```
dim(hogares)
```

```
## [1] 32664     3
```

```
table(hogares$region)
```

```
##
##    1    2    3    4    5    6    7    8    9   10   11   12   13   14   15
##  927  982  911 1960 4314 1983 2267 3347 1838 2702 1055  820 6240 1266 1055
```

Continuación profundización *dplyr*:

`if_else()` y `case_when()`

if_else()

La hemos ocupado para crear dos categorías:

```
esi <- esi %>%                              # Data
           mutate(filtro_mas_de_1M =           # Nueva variable
                    if_else(ing_t_d>1000000,  # Condición
                                         1,   # Verdadero
                                        0))   # Falso
```

```
table(esi$filtro_mas_de_1M)
```

```
##
##     0     1
## 92787  3453
```

¿Podríamos usar if_else() para crear variables con más de dos categorías?

if_else()

Pensemos en 4 tramos de ingresos...

Podemos usar varios if_else() consecutivamente

```
esi <- esi %>%

 mutate(filtro_4 =
  if_else(ing_t_d==0, "sin ingresos", ""),

 filtro_4 =
  if_else(ing_t_d>0 & ing_t_d<300000, "primer tramo", filtro_4),

 filtro_4 =
  if_else(ing_t_d<700000 & ing_t_d>=300000, "segundo tramo", filtro_4),

 filtro_4 =
  if_else(ing_t_d<1000000 & ing_t_d>=700000, "tercer tramo", filtro_4),

 filtro_4 =
  if_else(ing_t_d>=1000000, "cuarto tramo", filtro_4)
                  )
```

if_else()

```
table(esi$filtro_4)
```

```
##
##  cuarto tramo  primer tramo segundo tramo  sin ingresos  tercer tramo
##          3641          6674         15597         67011          3317
```

Funciona, pero es bastante código.

Hay otras formas de hacerlo más sencillo.

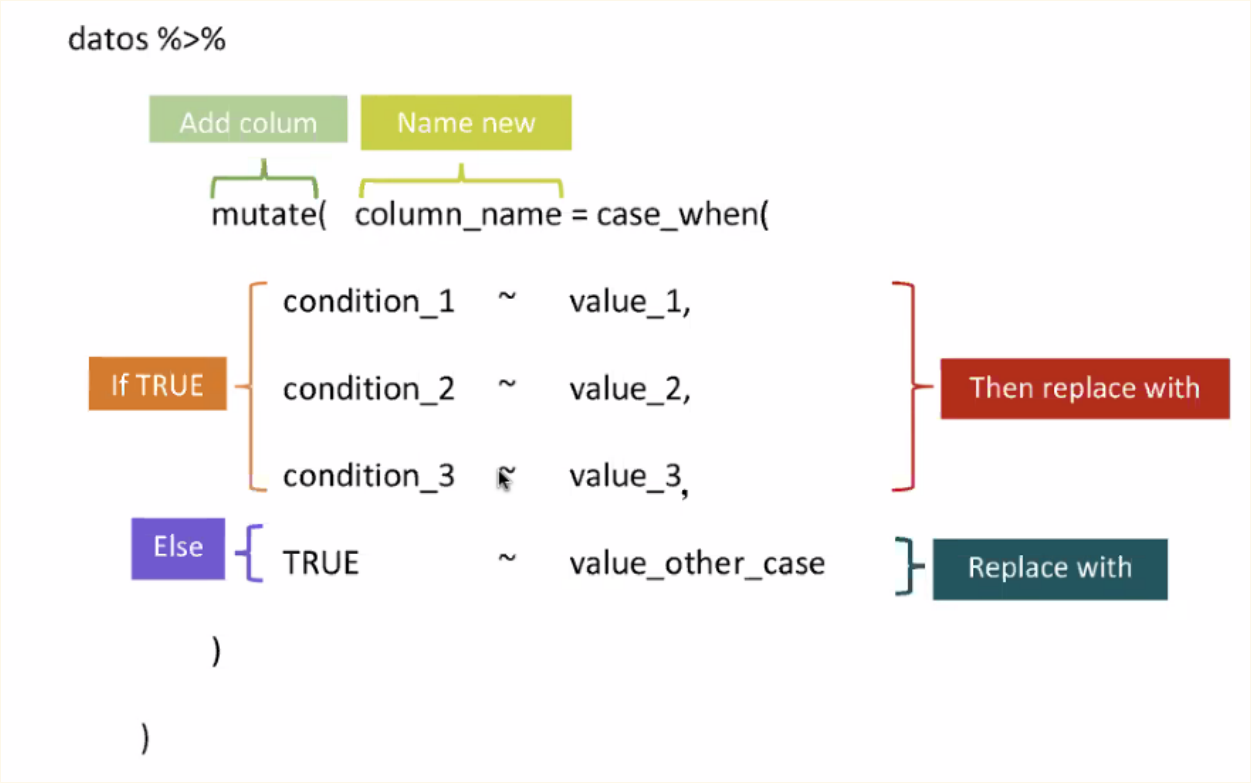
Una de esas es con case_when()

Esta es la lógica (cada línea independiente)

```
dataframe %>%
  mutate(nueva = case_when(condicion_1 ~ valor_1,
                           condicion_2 ~ valor_2,
                           condicion_3 ~ valor_3))
```

case_when()

Y así su forma general:



case_when()

Recodificar ingresos con case_when()

```
esi <- esi %>%
    mutate(filtro_4 = case_when(
              ing_t_d>0 & ing_t_d<300000 ~ "primer tramo",
              ing_t_d<700000 & ing_t_d>=300000 ~ "segundo tramo",
              ing_t_d<1000000 & ing_t_d>=700000 ~ "tercer tramo",
              ing_t_d>=1000000 ~ "cuarto tramo",
              TRUE ~ "sin ingresos",
                  ))
```

```
table(esi$filtro_4)
```

```
##
##  cuarto tramo  primer tramo segundo tramo  sin ingresos  tercer tramo
##          3641          6674         15597         67011          3317
```

case_when()

La virgulilla (~) tiene que se utilizada entre la condición y el valor a asignar.

`Alt + Control + + = ~`

Los valores a asignar deben ser de la misma clase.

No funciona con variables factores, tienen que ser *character* o *numeric*

La nueva variable debe ser numérica o carácter, no puede ser una combinación.

RECOMENDACIONES:

- Siempre probar la variable creada con `table( )` (que tenga sentido)
- Llamar con nuevo nombre a la nueva variable (no sobre escribir)
- Usar espacios para ordenar y facilitar la lectura

Paciencia con la función, al principio saltan varios errores

Prácticas para una buena codificación

Por Lindsay Carr

Por Lindsay Carr

- Carga de librerías al inicio del código (usando `library()`)
- Usa RStudio projects para organizar scripts, data y salidas
- Modulariza el código (todavía no)
- No guardar *workplace image*
- No usar funciones que cambian el computador de otro (`install.package()` o `setwd()`)
- Comenta el código, pero sin pasarte (no incluir interpretaciones o resultados).
- El principal destinatario de tus comentarios eres tú en 3 meses más.
- Si quieres interpretar el código y mostrar los resultados usa RMarkdown

Por Lindsay Carr

- Aprovecha el autollenado de RStudio (evita errores de tipeo)
- Copia y pega código utilizado anteriormente o por otros.
- Utiliza loops o **funciones** cuando te veas copiando y pegando código reemplazando valores
- Las tareas mecánicas en R pueden automatizarse
- Evita códigos anchos (sobre todo con pipes)

```
data %>%
  funcion1() %>%
  funcion2() %>%
  funcion3()
```

Recursos web utilizados

Xaringan: Presentation Ninja, de Yihui Xie. Para generar esta presentación.

Ilustraciones de Allison Horst

Para reforzar y seguir aprendiendo

Video "¿Cómo usar la función case_when en R? (tidyverse/dplyr)"

Buenas prácticas de codificación de Lindsay Carr

Bibliografía utilizada

Wickham, H. (2021). *R Para Ciencia de Datos*.