Solucion ejercicios sesión 3

Ejercicio 0

Lo primero es cargar los paquetes y los datos a utilizar.

```
library(tidyverse)
library(readr)
ene <- read_csv(file='data/ene-2019-11.csv')
```

Ejercicio 1

En este ejercio era posible usar `case_when` o `if_else`. Ambos caminos son igualmente correctos. Cabe mencionar que la condición para capturar los valores del intervalo [1,7] de *cae_especifico* puede escribirse de varias maneras. En este caso, se utilizó el operador `%in%`.

```
ene <- ene %>%
  mutate(pet = if_else(edad >= 15, 1, 0),
         ocu = if_else(cae_especifico %in% 1:7, 1, 0))
```

Ejercicio 2

Lo primero era agrupar por región y luego sumar separadamente la ocupación (ocu) y la población en edad de trabajar (pet). Es importante recordar siempre desagrupar.

```
tabla_region <- ene %>%
  group_by(region) %>%
  summarise(suma_ocu = sum(ocu), suma_pet = sum(pet)) %>%
  ungroup() # siempre desagrupar
```

Ejercicio 3

Usamos el objeto creado en el ejercicio anterior y calculamos la tasa de ocupación para cada una de las regiones, mediante la función `mutate`

```
tabla <- tabla_region %>%
  mutate(tasa_ocup = (suma_ocu / suma_pet) * 100)
```

En la vida real, usualmente, los pasos anteriores se llevan a cabo de una sola vez, ya que ello evita generar objetos intermedios que no serán utilizados y que ocupan espacio en memoria innecesariamente. Adicionalmente, es posible acortar un paso y generar el valor deseado dentro de `summarise`

```
ene %>%
  mutate(pet = if_else(edad >= 15, 1, 0),
         ocu = if_else(cae_especifico %in% 1:7, 1, 0)) %>%
  group_by(region) %>%
  summarise(tasa_ocup = (sum(ocu) / sum(pet)) * 100)
```

```
## # A tibble: 16 x 2
##    region tasa_ocup
##     <dbl>     <dbl>
```

```
##  1      1      56.5
##  2      2      57.3
##  3      3      56.9
##  4      4      55.7
##  5      5      52.8
##  6      6      55.0
##  7      7      55.1
##  8      8      48.6
##  9      9      53.0
## 10     10      53.9
## 11     11      66.6
## 12     12      62.3
## 13     13      56.8
## 14     14      56.4
## 15     15      56.9
## 16     16      52.2
```

Ejercicio 4

Este ejercicio se divide en dos partes. En primer lugar, es necesario generar un conteo de filas agrupadas por las dos variables solicitadas (*b14_rev4cl_caenes* y *b1*). Esto se logra mediante las funciones `group_by` y `summarise`. El resultado es una columna llamada *n*, que contiene el número de filas para cada combinación de *b14_rev4cl_caenes - b1*

La segunda parte corresponde al "pivoteo". En este caso, queremos que los datos de *b1* estén ordenados en columnas. Para ello, indicamos que los nombres de las columnas serán obtenidos de *b1* y los valores, de *n* (creada en el paso anterior). El uso del parámetro *names_prefix* es opcional. Este permite nombrar a las columnas nuevas de manera sencilla.

```
tabla_wide <-  ene %>%
  group_by(b14_rev4cl_caenes, b1) %>%
  summarise(n = n()) %>%
  pivot_wider(names_from = b1,
              names_prefix = "grupo",
              values_from = n)
```