Respuestas Tarea 1

Code

Tarea 1

Taller de Análisis de Datos 1 Enzo Loiza

Hide

```
gc()
```

```
         used (Mb) gc trigger  (Mb) max used  (Mb)
Ncells 1185219 63.3    2053565 109.7  2053565 109.7
Vcells 2719150 20.8   10146329  77.5  7025327  53.6
```

Hide

```
rm(list=ls())

# use libraries
library(tidyverse)
library(readxl)
```

Ejercicio 1

a. Generamos en el `environment` los siguientes vectores que se piden:

Hide

```
vector_1 <- seq(1, 5, .5)
vector_2 <- seq(1, 20, 2)
vector_3 <- replicate(10, 2023)
vector_4 <- as.numeric(cbind(replicate(5,1),replicate(5,0)))
vector_5 <- vector_1 + vector_2
```

```
Warning: longer object length is not a multiple of shorter object length
```

Los mostramos a continuación:

Hide

```
vector_1
```

```
[1] 1.0 1.5 2.0 2.5 3.0 3.5 4.0 4.5 5.0
```

Hide

```
vector_2
```

```
 [1]  1  3  5  7  9 11 13 15 17 19
```

Hide

```
vector_3
```

```
 [1] 2023 2023 2023 2023 2023 2023 2023 2023 2023 2023
```

Hide

```
vector_4
```

```
 [1] 1 1 1 1 1 0 0 0 0 0
```

Hide

```
vector_5
```

```
 [1]  2.0  4.5  7.0  9.5 12.0 14.5 17.0 19.5 22.0 20.0
```

Los vectores `vector_1` y `vector_2` tienen distinta longitud (9 y 10 respectivamente). Por lo tanto, la coerción para la suma de ambos vectores corresponde a que se suman los últimos valores del `vector_2` (el vector más largo) con los primeros del `vector_1` (el vector más corto), y el vector final quedará con la longitud del más largo. En este caso específico, el `vector_5` tendrá longitud 10.

Explicamos la diferencia entre los vectores, matrices y dataframes: - `Vector` : es una colección de datos que pueden ser sólo de un mismo tipo. Su única dimensión es el largo. Los números son en sí vectores de largo 1. Ejemplos:

```
v <- 1
a <- c(1,2)
b <- seq(1,10)
```

- `Matrix` : las matrices son colecciones de vectores, estructuras de dos dimensiones que contienen vectores, todos del mismo largo. Además del largo, tienen ancho, pues se ordenan de manera matricial con las mismas características de las matrices algebraicas. Ejemplos:

  ```
  B <- matrix(1:9)
  A <- matrix(1:12, nrow = 5, ncol = 4)
  ```

- `DataFrame` : las dataframes son estructuras de dos dimensiones donde cada vector puede contener un tipo distinto de dato. Son versiones más flexibles de las matrices.

b. Generamos la matriz que se nos solicita, usando los vectores de la parte a, y la mostramos:

Hide

```
M <- cbind(vector_1,
           vector_2,
           vector_3,
           vector_4,
           vector_5)
```

```
Warning: number of rows of result is not a multiple of vector length (arg 1)
```

Hide

```
M
```

```
      vector_1 vector_2 vector_3 vector_4 vector_5
 [1,]      1.0        1     2023        1      2.0
 [2,]      1.5        3     2023        1      4.5
 [3,]      2.0        5     2023        1      7.0
 [4,]      2.5        7     2023        1      9.5
 [5,]      3.0        9     2023        1     12.0
 [6,]      3.5       11     2023        0     14.5
 [7,]      4.0       13     2023        0     17.0
 [8,]      4.5       15     2023        0     19.5
 [9,]      5.0       17     2023        0     22.0
[10,]      1.0       19     2023        0     20.0
```

El `vector_1` tiene un largo menor a los demás vectores, por lo tanto, la coerción que se aplica a este vector corresponde a: 1. establecer el largo de la matriz como el largo mayor de los vectores, y 2. reciclar los valores de los vectores con largo menor. Esto nos importa para entender qué pasa al trabajar con los datos.

La diferencia está en el tipo de datos que queremos manejar. Si necesitamos trabajar únicamente con datos numéricos, como al hacer una regresión lineal, una matriz es suficiente. Sin embargo, si queremos ver diferencias entre variables tratando los datos y creando variables binarias, necesitaremos trabajar con dataframes.

c. Primero extraemos la base de datos para trabajarla

Hide

```
simce <- read.csv("simce2m2016_extracto.csv", sep = ',', dec = '.')
names(simce)
```

```
[1] "idalumno"        "gen_alu"         "ptje_mate2m_alu" "ptje_lect2m_alu"
[5] "cpad_p08"        "cod_depe2"
```

Los datos de simce que nos interesan para hacer la evaluación son `ptje_lect2m_alu` (puntaje lenguage) y `ptje_mate2m_alu` (puntaje matemáticas), por lo que usaremos estas variables para hacer la regresión.

Hide

```
regresion <- lm(ptje_mate2m_alu ~ ptje_lect2m_alu,
                data = simce,
                na.rm = TRUE)
```

```
Warning: In lm.fit(x, y, offset = offset, singular.ok = singular.ok, ...) :
 extra argument ‘na.rm’ will be disregarded
```

Hide

```
regresion
```

```
Call:
lm(formula = ptje_mate2m_alu ~ ptje_lect2m_alu, data = simce,
    na.rm = TRUE)

Coefficients:
    (Intercept)  ptje_lect2m_alu
        78.9131           0.7581
```

Podemos ver que una regresión lineal simple nos idica que por cada punto de matemáticas, aumenta 0.7581 puntos en la prueba de lenguaje en promedio. La afirmación del enunciado es falsa, pues el β de la regresión es positivo. Podemos hacer doble click en la regresión pidiendo un resumen de ésta:

Hide

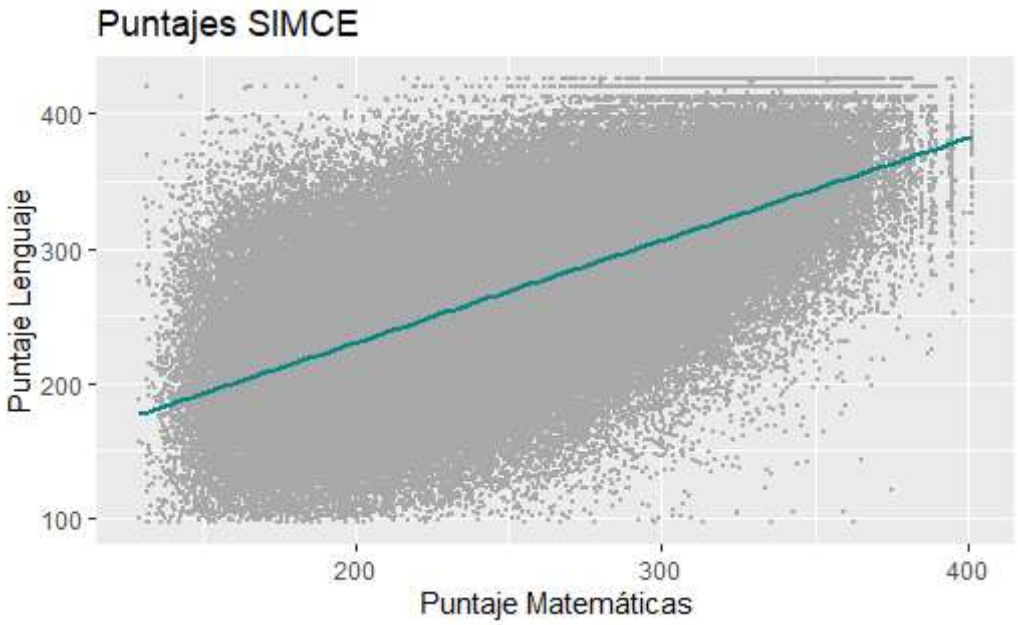
```
summary(regresion)
```

```
Call:
lm(formula = ptje_mate2m_alu ~ ptje_lect2m_alu, data = simce,
    na.rm = TRUE)

Residuals:
     Min       1Q   Median       3Q      Max
-255.973  -35.029    0.472   35.174  241.423

Coefficients:
                 Estimate Std. Error t value Pr(>|t|)
(Intercept)     78.913080   0.628065   125.6   <2e-16 ***
ptje_lect2m_alu  0.758091   0.002464   307.7   <2e-16 ***
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 51.44 on 160177 degrees of freedom
Multiple R-squared:  0.3714,    Adjusted R-squared:  0.3714
F-statistic: 9.465e+04 on 1 and 160177 DF,  p-value: < 2.2e-16
```

Con lo que obtenemos errores bastante grandes, aunque la tendencia existe. Podemos verificar esto a través de un gráfico,

Hide

```
ggplot(simce,
       aes(ptje_lect2m_alu,
           ptje_mate2m_alu)) +
  geom_point(size = .3,
             color = 'darkgrey')+
  geom_smooth(method='lm', color='turquoise4') +
  labs(x='Puntaje Matemáticas',
       y='Puntaje Lenguaje',
       title='Puntajes SIMCE')
```



Pregunta 2

Primero abrimos la librería:

Hide

```
library(WDI)
```

a. Elegimos un indicador mediante una búsqueda a través del comando `WDIsearch()`. El indicador elegido corresponde a la esperanza de vida al nacer. Este valor tiene mucha importancia tanto en salud como en seguridad social (al establecer, por ejemplo, políticas de pensión).

Hide

```
WDIsearch("life expectancy at birth.*total")
```

	indicator <chr>	name <chr>
17308	SP.DYN.LE00.IN	Life expectancy at birth, total (years)

1 row

El indicador se llama `SP.DYN.LE00.IN`, pero en nuestra base de datos (`DB`) la llamaremos `leab`.

Hide

```
DB <- WDI(country = 'all',
          indicator = c("leab"="SP.DYN.LE00.IN"),
          start = 2015,
          end = 2020,
          extra = TRUE,
          cache = NULL,
          latest = NULL,
          language = 'en') %>%
  filter(region == 'Latin America & Caribbean')
```

Vemos un pequeño resumen de la base obtenida:

Hide

```
summary(DB)
```

```
  country             iso2c              iso3c                year     
Length:252         Length:252         Length:252         Min.   :2015  
Class :character   Class :character   Class :character   1st Qu.:2016  
Mode  :character   Mode  :character   Mode  :character   Median :2018  
                                                         Mean   :2018  
                                                         3rd Qu.:2019  
                                                         Max.   :2020  

     leab          status          lastupdated          region         
Min.   :63.24   Length:252         Length:252         Length:252        
1st Qu.:72.57   Class :character   Class :character   Class :character  
Median :74.41   Mode  :character   Mode  :character   Mode  :character  
Mean   :74.54                                                           
3rd Qu.:77.08                                                           
Max.   :80.61                                                           
NA's   :9                                                               
  capital           longitude           latitude            income         
Length:252         Length:252         Length:252         Length:252        
Class :character   Class :character   Class :character   Class :character  
Mode  :character   Mode  :character   Mode  :character   Mode  :character  



  lending         
Length:252        
Class :character  
Mode  :character  
```

Que nos indica que tenemos 252 observaciones. Luego,

Hide

```
DB %>% count(year) # numero de años 6
```

year <int>	n <int>
2015	42
2016	42
2017	42
2018	42
2019	42
2020	42

6 rows

Hide

```
DB %>% count(country) %>% summary() # numero de paises 42
```

```
   country                n         
 Length:42          Min.   :6  
 Class :character   1st Qu.:6  
 Mode  :character   Median :6  
                    Mean   :6  
                    3rd Qu.:6  
                    Max.   :6  
```

Vemos que la data contiene 6 años desde 2015 hasta 2020, y 42 países.

b. Hacemos la tabla resumen de los valores de esperanza de vida por año.

Hide

```
tabla <- DB %>%
  group_by(year) %>%
  summarise(na.rm = TRUE,
    N = sum(!is.na(leab)),
    mean = mean(leab, na.rm = TRUE),
    sd = sd(leab, na.rm = TRUE),
    min = min(leab, na.rm = TRUE),
    median = median(leab, na.rm = TRUE),
    max = max(leab, na.rm = TRUE)
  ) %>%
  left_join(DB %>% distinct(year, region), by = "year")

tabla[,c('region', 'year', 'N', 'mean', 'sd', 'min','median', 'max')]
```

region <chr>	year <int>	N <int>	mean <dbl>	sd <dbl>	min <dbl>	median <dbl>	max <dbl>
Latin America & Caribbean	2015	41	74.51728	3.608612	63.237	74.6820	79.746
Latin America & Caribbean	2016	41	74.63539	3.530230	63.392	74.4420	80.270
Latin America & Caribbean	2017	41	74.58835	3.516664	63.854	74.7650	80.350
Latin America & Caribbean	2018	40	74.79784	3.417537	64.019	74.4680	80.614
Latin America & Caribbean	2019	40	74.75241	3.396188	64.255	74.5455	80.326
Latin America & Caribbean	2020	40	73.94311	3.669186	64.052	73.6570	80.149

6 rows

La esperanza de vida se mantuvo más o menos constante hasta 2019. Sin embargo, en 2020, el indicador presentó una pequeña variación negativa. La desviación estándar, por otro lado, fue disminuyendo decimalmente los primeros 5 años, pero aumentó levemente en 2020. Estos resultados pueden deberse a la crisis sanitaria mundial producto del coronavirus.

c. Veremos si tenemos missing values. Simplemente de la tabla anterior, sabemos que existen, ahora vamos a encontrarlos:

Hide

```
DB %>%
  filter(is.na(leab)) %>%
  select(country, year)
```

country <chr>	year <int>
Cayman Islands	2019
Cayman Islands	2018
Cayman Islands	2020
Cayman Islands	2016
Cayman Islands	2015
Cayman Islands	2017
Curacao	2019
Curacao	2018
Curacao	2020

9 rows

Es decir, existen missing values para las Islas Cayman (en los 6 años de análisis) y Curacao (desde 2018).

Podemos establecer varias estrategias pero dos son interesantes:

- Para los dos casos con missing data, podemos asumir suavidad entre países vecinos. Es decir, por ejemplo, la esperanza de vida de Paraguay, no es muy diferente a la de sus vecinos (Argentina, Bolivia, Brasil). Por lo que para cada año, podemos hacer una polarización ponderada desde los vecinos, usando los valores que sí existen en la data.
- Una segunda opción válida para Curacao, es utilizar los valores que sí existen (2015 y 2016) y extrapolarlos a partir de la información que tenemos de las tendencias de esperanza de vida en Latinoamérica y Caribe.
- Probablemente la mejor opción que tenemos es una mezcla de ambas estrategias.

d. A nuestra base ahora le haremos un filtro para chile:

Hide

```
chile <- DB %>% filter(country =="Chile")
```

```
Error in exists(cacheKey, where = .rs.WorkingDataEnv, inherits = FALSE) :
  invalid first argument
Error in assign(cacheKey, frame, .rs.CachedDataEnv) :
  attempt to use zero-length variable name
```

Resumimos igual que en el punto b., es decir haciendo una tabla. Esta vez incluiremos también un gráfico para mostrar de mejor forma.

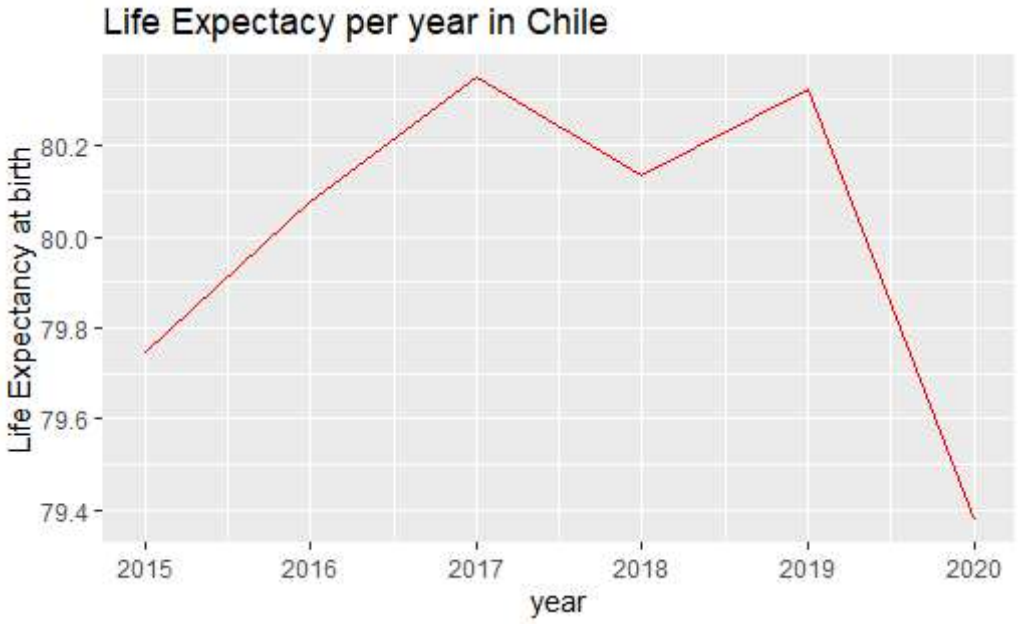
Hide

```
summary(chile$leab)
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
  79.38   79.83   80.11   80.00   80.28   80.35 
```

Hide

```
ggplot(data = chile, aes(year, leab)) +
  geom_line(colour = 'red') +
  xlab('year') +
  ylab('Life Expectancy at birth') +
  ggtitle('Life Expectacy per year in Chile')
```



La esperanza de vida al nacer muestra una tendencia positiva en los años 2015-2019, sin embargo, tiene una caída brusca en 2020, posiblemente (al igual que los otros países de la región) debido a la crisis sanitaria.

e. Puedo resumir mis análisis en los siguientes puntos:

- En toda la región se muestra un aumento sostenido de la espezanza de vida al nacer desde 2015 al 2019.
- El aumento sostenido se ve paralizado en 2020 debido a la pandemia que azotó tanto a la región como al mundo entero.
- Este aumento es posible que se recupere si se toman medidas de política pública en salud pues responde a mejoras en el servicio médico.
- Por otro lado, el aumento también tiene consecuencias a largo plazo en relación con las políticas de seguridad social en la vejez. Una mala gestión de recursos puede favorecer un colapso del sistema de pensiones que deje a una gran parte de la población de tercera edad de un país sin recursos para sus manutenciones.

Bonus

Primero instalamos la librería necesaria

Hide

```
library(sf)
```

Luego, cargamos el mapa de Latinoamérica:

Hide

```
latam_shape <- ne_countries(scale = "medium",
                            type = "map_units",
                            country = "countries",
                            continent = c("South America",
                                          "North America"))
```

No creo que alcance a hacerla :(