MetII_T1_MEF

Daniel Carrasco

```
# Defino mi la ruta de mi archivo
setwd("G:/Mi unidad/R a Python Proy. MEF/Tarea Met II/Sol en R T1")
getwd()
```

```
[1] "G:/Mi unidad/R a Python Proy. MEF/Tarea Met II/Sol en R T1"
```

```
# Cargo la CASEN
load("CASEN2013.RData")
```

Análisis de exploratorio para escolaridad de encuesta CASEN 2013

- Usando la CASEN 2013, crearemos un data.frame llamado "edadEscolaridad" con las siguientes variables: "folio", "o", "pco1", "region", "sexo", "expr", "edad", "ESC".
- Debemos definir cada una de las varibles con rigurosidad.

```
# comprobar que las variables existen en data00
a <- which(names(data00)%in%c( "folio", "o",
                           "pco1", "region",
                           "sexo", "expr",
                           "edad", "ESC"))

names(data00[a])
```

```
[1] "folio"  "o"      "region" "pco1"   "sexo"   "edad"   "expr"   "ESC"
```

```r
#crear dataframe

edadEscolaridad <- data00[c( "folio", "o",
                             "pco1", "region",
                             "sexo", "expr",
                             "edad", "ESC")]
summary(edadEscolaridad)
```

```
     folio                  o               pco1            region
 Min.   :1.101e+10   Min.   : 1.000   Min.   : 1.000   Min.   : 1.000
 1st Qu.:5.802e+10   1st Qu.: 1.000   1st Qu.: 1.000   1st Qu.: 5.000
 Median :8.307e+10   Median : 2.000   Median : 3.000   Median : 8.000
 Mean   :8.610e+10   Mean   : 2.544   Mean   : 3.236   Mean   : 8.407
 3rd Qu.:1.310e+11   3rd Qu.: 3.000   3rd Qu.: 4.000   3rd Qu.:13.000
 Max.   :1.520e+11   Max.   :19.000   Max.   :14.000   Max.   :15.000

      sexo            expr              edad            ESC
 Min.   :1.000   Min.   :   2.00   Min.   :  0.00   Min.   : 0.00
 1st Qu.:1.000   1st Qu.:  28.00   1st Qu.: 17.00   1st Qu.: 8.00
 Median :2.000   Median :  50.00   Median : 33.00   Median :12.00
 Mean   :1.523   Mean   :  79.06   Mean   : 35.45   Mean   :10.29
 3rd Qu.:2.000   3rd Qu.:  89.00   3rd Qu.: 52.00   3rd Qu.:12.00
 Max.   :2.000   Max.   :6812.00   Max.   :108.00   Max.   :22.00
                                                    NA's   :46885
```

Como ya tenemos el data frame buscado, podemos eliminar los datos originales para mejorar el procesamiento.

```r
rm(data00) #rm = remove, permite eliminar objetos
```

Definimos cada una de las variables

- o = Numero de orden de la persona dentro del hogar.
- folio = Identificación del hogar.
- region = Región
  - 1 I: Tarapaca
  - 2 II: Antofagasta
  - 3 III: Atacama

  - 4 IV: Coquimbo
  - 5 V: Valparaíso
  - 6 VI: Libertador General Bernardo O'Higgins
  - 7 VII: Maule
  - 8 VIII: Bío Bío
  - 9 IX: La Araucanía
  - 10 X: Los Lagos
  - 11 XI: Aysén del General Carlos Ibáñez del Campo
  - 12 XII: Magallanes y de la Antírtica Chilena
  - 13 R.M.: Metropolitana de Santiago
  - 14 XIV: Los Ríos
  - 15 XV: Arica y Parinacota
- pco1 = Parentesco con el jefe de hogar
  - 1 Jefe(a) de hogar
  - 2 Esposo(a) o pareja
  - 3 Hijo(a) de ambos
  - 4 Hijo(a) sólo del jefe
  - 5 Hijo(a) sólo del esposo(a) o pareja
  - 6 Padre o madre
  - 7 Suegro(a)
  - 8 Yerno o nuera
  - 9 Nieto(a)
  - 10 Hermano(a)
  - 11 Cuñado(a)
  - 12 Otro familiar
  - 13 No familiar
  - 14 Servicio doméstico puertas adentro
- sexo

  - 1 hombre
  - 2 mujer

- expr = expanasión regional
- edad = edad
- ESC = escolaridad nivel de educación

Usando la función "apply", calcularemos el promedio para edad y escolaridad de los/as jefes/as de hogar solamente y sin remover los NA.

- Utilizaremos el comando Apply para derermina la media sobre todo el data frame filtrado por la condición indicada
- con el "which" filtramos el data frame seleccionando solo las que tengan el atributo de jefe de hogar (pco1 = 1)
- De ese filtro generado tomamos las columnas "edad" y "ESC" y le aplicamos la media

```
apply(edadEscolaridad[which(edadEscolaridad$pco1==1),c("edad","ESC")],2,mean)
```

```
    edad      ESC
52.79706       NA
```

Como observamos para la columna escolaridad nos dio un resultado "NA".

Esto es debido a que la aplicación de la media en la columna no se ejecutó correctamente, el error basicamentes es por que en la columnas existes valores NA, de los cuales no se puede determinar una media

Con la misma función determinaremos las medias quitandos los NAs

```
apply(edadEscolaridad[which(edadEscolaridad$pco1==1),c("edad","ESC")],2,mean, na.rm=TRUE)
```

```
     edad       ESC
52.797063  9.606913
```

El comando "na.rm" remueve los NA de las columnas indicadas

Con esto hemos quitado las filas en las cuales habia un NA en uno de las dos columnas.

Con esto sabemos que la edad promedio de escolaridad de los jefes de hogar es de 9,6 años. Una posible conclusión podria ser que en el 2013 los jefes de hogar en Chile tenia en promedio la ensañanza basica completa.

Función Apply

La función `apply()` en R se utiliza para **realizar operaciones repetitivas** sobre las **filas o columnas** de una **matriz** o un `data.frame` numérico. Es una forma eficiente de aplicar una función (como promedio, suma, máximo, etc.) sin necesidad de usar bucles (`for`, `while`, etc.).

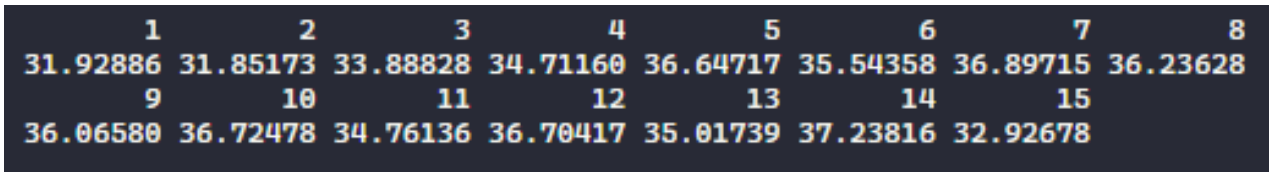
Función Tapply

La función `tapply()` en R se utiliza para **aplicar una función a subconjuntos de un vector**, definidos por una o más variables categóricas (factores). Es ideal para **hacer resúmenes agrupados**, como calcular promedios, sumas o conteos por grupo.

Usaremos la función Tapply para determinar el promedio de años de edad para cada región

```
tapply(edadEscolaridad$edad, edadEscolaridad$region, mean)
```

```
       1        2        3        4        5        6        7        8
31.92886 31.85173 33.88828 34.71160 36.64717 35.54358 36.89715 36.23628
       9       10       11       12       13       14       15
36.06580 36.72478 34.76136 36.70417 35.01739 37.23816 32.92678
```



Ahora determinaremos el promedio de años de escolaridad por región utilizando la misma función

```
tapply(edadEscolaridad$ESC, edadEscolaridad$region, mean)
```

```
 1  2  3  4  5  6  7  8  9 10 11 12 13 14 15
NA NA NA NA NA NA NA NA NA NA NA NA NA NA NA
```

Como se observa todas las regiones poseen datos perdidos o NAs, removeremos estos registros.

```
as.data.frame.table(
  tapply
  (edadEscolaridad$ESC,edadEscolaridad$region, mean, na.rm=TRUE))
```

```
   Var1      Freq
1     1 10.995247
2     2 11.281495
3     3 10.563376
4     4 10.253363
5     5 10.779830
6     6  9.801758
7     7  9.188465
8     8  9.967137
9     9  9.622209
10   10  9.307767
11   11  9.778033
12   12 11.030174
13   13 11.190112
14   14  9.643455
15   15 10.948714
```

Opción grafica diferente

```
library(DT)

DT::datatable(
  as.data.frame.table(
    tapply(edadEscolaridad$ESC,
           INDEX = edadEscolaridad[c("sexo", "region")],
           mean,
           na.rm = TRUE)
  ),
  options = list(pageLength = 20),
  caption = "Promedio de escolaridad por sexo y región"
)
```

file:///C:/Users/Daniel/AppData/Local/Temp/Rtmp08ICTx/file21d44f4520a5/widget21d463fd14c1.htı

Show 20 entries Search:

Promedio de escolaridad por sexo y región

	sexo	region	Freq
1	1	1	11.14007447722716
2	2	1	10.86470436354247
3	1	2	11.45796610169491
4	2	2	11.1176840780365
5	1	3	10.7789046653144
6	2	3	10.36289308176101
7	1	4	10.31572029442692
8	2	4	10.19913122999543
9	1	5	10.90658398205568
10	2	5	10.67093629562457
11	1	6	9.788953676708779
12	2	6	9.813392200147167
13	1	7	9.077891280554542
14	2	7	9.286709886547811
15	1	8	10.05247687386531
16	2	8	9.893388121031006
17	1	9	9.651133501259446
18	2	9	9.59691629955947
19	1	10	9.457055214723926
20	2	10	9.174429223744292

Showing 1 to 20 of 30 entries Previous 1 2 Next

Utilizando la misma función determinaremos los años promedios de escolaridad para cada región aperturando por sexo.

```
as.data.frame.table(tapply(
  edadEscolaridad$ESC,INDEX =  edadEscolaridad[c("sexo","region")], mean,
  na.rm=TRUE))
```

```
   sexo region      Freq
1     1      1 11.140074
2     2      1 10.864704
3     1      2 11.457966
4     2      2 11.117684
5     1      3 10.778905
6     2      3 10.362893
7     1      4 10.315720
8     2      4 10.199131
9     1      5 10.906584
10    2      5 10.670936
11    1      6  9.788954
12    2      6  9.813392
13    1      7  9.077891
14    2      7  9.286710
15    1      8 10.052477
16    2      8  9.893388
17    1      9  9.651134
18    2      9  9.596916
19    1     10  9.457055
20    2     10  9.174429
21    1     11  9.848644
22    2     11  9.712389
23    1     12 11.201417
24    2     12 10.872088
25    1     13 11.387773
26    2     13 11.020277
27    1     14  9.720909
28    2     14  9.573845
29    1     15 11.117021
30    2     15 10.803123
```

Hint: explicación del comando

- *edadEscolaridad$ESC : Es el* ***vector numérico*** *al que se le aplicará la función* **`mean`** *(el nivel de escolaridad)*

- *INDEX = edadEscolaridad[c("sexo", "region")] : Define los* ***grupos*** *por los cuales se quiere agrupar, en este caso, combinando las columnas* `sexo` *y* `region`
- *mean : Es la* ***función*** *que se aplicará a cada grupo*

```
as.data.frame.table(
  tapply(edadEscolaridad$ESC,
         INDEX = list(sexo = factor(edadEscolaridad$sexo, levels = c(1, 2), labels = c("Hombr
                      region = edadEscolaridad$region),
         mean,
         na.rm = TRUE)
)
```

```
     sexo region      Freq
1  Hombre      1 11.140074
2   Mujer      1 10.864704
3  Hombre      2 11.457966
4   Mujer      2 11.117684
5  Hombre      3 10.778905
6   Mujer      3 10.362893
7  Hombre      4 10.315720
8   Mujer      4 10.199131
9  Hombre      5 10.906584
10  Mujer      5 10.670936
11 Hombre      6  9.788954
12  Mujer      6  9.813392
13 Hombre      7  9.077891
14  Mujer      7  9.286710
15 Hombre      8 10.052477
16  Mujer      8  9.893388
17 Hombre      9  9.651134
18  Mujer      9  9.596916
19 Hombre     10  9.457055
20  Mujer     10  9.174429
21 Hombre     11  9.848644
22  Mujer     11  9.712389
23 Hombre     12 11.201417
24  Mujer     12 10.872088
25 Hombre     13 11.387773
26  Mujer     13 11.020277
27 Hombre     14  9.720909
28  Mujer     14  9.573845
29 Hombre     15 11.117021
```

```
30  Mujer     15 10.803123
```

Como podemos determinar en que región existe mayor brecha de escolaridad?

```
which.max(
  abs(
    tapply(edadEscolaridad$ESC,INDEX =  edadEscolaridad[c("sexo","region")], mean, na.rm=TRU
  tapply(edadEscolaridad$ESC,INDEX =  edadEscolaridad[c("sexo","region")], mean, na.rm=TRUE)
))
```

```
3
3
```

Hint: explicación del comando

- *tapply(…)[1,]:*
  - *Calcula el* ***promedio de escolaridad*** *(ESC) para* ***sexo 1*** *(hombres), por región.*
  - *Devuelve una* ***fila con un valor promedio por cada región****.*
- tapply(…)[2,] : *Hace lo mismo que el anterior pero para las mujeres*
- *[1,] - [2,] : Esto calcula la* ***diferencia en escolaridad entre hombres y mujeres*** *para cada región.*
- *abs(…) : Aplica el valor absoluto a cada diferencia. Así se ignora si la diferencia es a favor de hombres o mujeres, y se enfoca solo en* ***la magnitud de la brecha***
- *which.max(…) : Devuelve el* ***índice (posición)*** *de la región donde la* ***diferencia absoluta entre hombres y mujeres es mayor****.*

Usando datos expandidos

Se solicita crear un dataframe llamado "edadEscolaridad pco1" el cual incluye solamente al grupo de jefes de hogar, además de considerar igualmente las columnas solicitadas en el primer ejercicio.

```
edadEscolaridadpc1 <- edadEscolaridad[which(edadEscolaridad$pco1==1),]
```

Ahora se solicita lo siguiente "Usando la función "apply", "tapply" y "weighted.mean" en conjunto, calcule el promedio para edad y escolaridad removiendo los NA por región. Debe crear/usar una función propia para lograr esto. Guarde los resultados en un objeto llamado "edadYEscPorRegExp" "

```r
edadYEscPorRegExp <- apply(edadEscolaridadpc1[c("edad","ESC")],2,
      function(x){ ind <- seq(along=x)
      tapply(ind,INDEX = edadEscolaridadpc1$region,
             function(i) weighted.mean(x[i],edadEscolaridadpc1$expr[i],na.rm = T)
      )}

)
edadYEscPorRegExp
```

```
       edad       ESC
1  49.07949 10.907801
2  48.61091 11.395684
3  50.95613  9.942783
4  52.94196  9.718694
5  53.47106 10.740477
6  51.99387  9.433958
7  53.68287  8.294571
8  53.70267  9.343049
9  53.34361  8.951130
10 52.87064  8.926648
11 50.78826  9.303072
12 52.52604 10.573716
13 51.84211 11.239140
14 55.36900  8.817700
15 51.18479 10.429524
```

Hint: explicación del comando

- *apply(..., 2, function(x) {...}) :*
  - ***`apply()`*** *se usa para aplicar una función sobre* ***las columnas*** *(`MARGIN = 2`) de un data frame o matriz. En este caso se está aplicando sobre las columnas* ***`"edad"`*** *y* ***`"ESC"`*** *del data frame* ***`edadEscolaridadpc1`****.*
- *function(x) :*
  - *La función interna toma una columna (**`x`**), que puede ser* ***`"edad"`*** *o* ***`"ESC"`*** *según el momento.*
- ind <- seq(along = x) :
  - Crea un índice secuencial: `1:length(x)`, necesario para aplicar `tapply` correctamente.

- tapply(ind, INDEX = edadEscolaridadpc1$region, …) :
  - Agrupa los datos por región (`region`), y aplica una función a cada grupo.
  - Dentro de cada grupo (`i`), calcula un **promedio ponderado** de los valores `x[i]`.
- weighted.mean(x[i], edadEscolaridadpc1$expr[i], na.rm = TRUE) :
  - Aquí se calcula el promedio ponderado de los valores `x[i]` (edad o escolaridad), usando como **peso** la columna `expr[i]`.

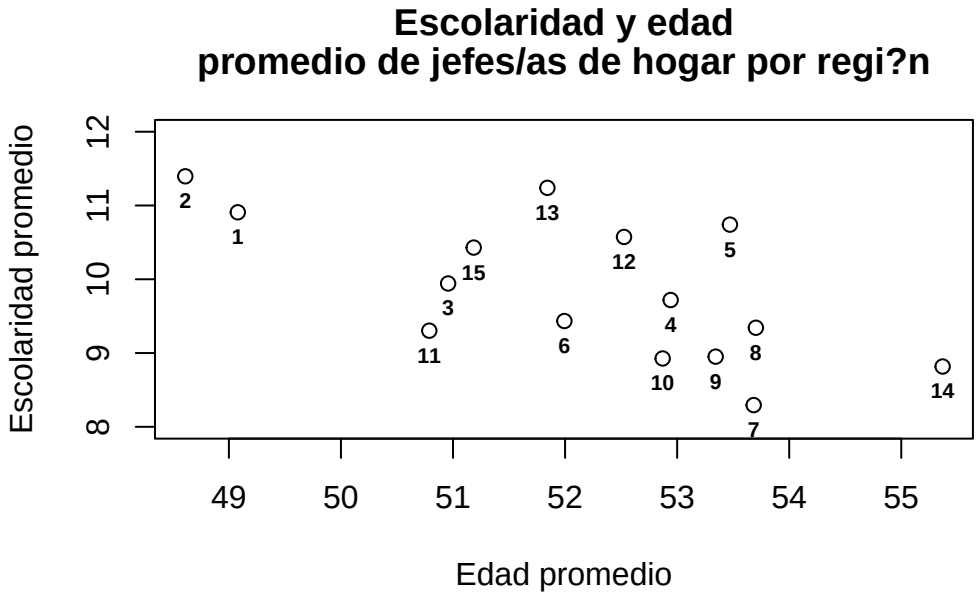
clave:

Elemento del código	¿Qué es?	¿Qué hace?	Ejemplo simple
`x`	Vector de datos (`edad` o `ESC`)	Los valores sobre los que se quiere calcular el promedio ponderado	`x = c(25, 30, 28, 40, 23)`
`expr`	Vector de pesos	Factor de expansión para cada observación	`expr = c(1.2, 0.8, 1.5, 1.1, 0.9)`
`ind <- seq(along = x)`	Índices del vector `x`	Crea una secuencia 1, 2, 3,... de la misma longitud que `x`	`ind = 1:5`
`INDEX = edadEscolaridadpc1$region`	Agrupador	Determina cómo agrupar los datos (por región)	`region = c(1,1,2,2,1)`
`tapply(ind, INDEX, ...)`	Agrupación	Agrupa los índices `ind` según `region`	Región 1: `i = c(1,2,5)` Región 2: `i = c(3,4)`
`function(i)`	Función para aplicar	Para cada grupo `i`, aplica un cálculo	`i` = posiciones del grupo actual
`x[i]`	Datos seleccionados	Extrae los valores `x` del grupo actual	Si `i = c(1,2,5)`, `x[i] = c(25,30,23)`
`expr[i]`	Pesos seleccionados	Extrae los pesos correspondientes a `x[i]`	`expr[i] = c(1.2, 0.8, 0.9)`
`weighted.mean(x[i], expr[i])`	Promedio ponderado	Calcula el promedio ponderado del grupo actual	Ejemplo: `(25×1.2 + 30×0.8 + 23×0.9) / (1.2+0.8+0.9)`

Generamos un grafico de dispersión

```
plot(edadYEscPorRegExp, ylim=c(8,12),
     main = c("Escolaridad y edad","promedio de jefes/as de hogar por regi?n"),
     xlab = "Edad promedio", ylab = "Escolaridad promedio")

text(edadYEscPorRegExp[,1],edadYEscPorRegExp[,2],
     labels = row.names(edadYEscPorRegExp) ,
     cex = .7,adj = c(0,0),pos = 1, font = 2)
```




```
plot(edadYEscPorRegExp, ylim=c(8,12),
     main = c("Escolaridad y edad","promedio de jefes/as de hogar por regi?n"),
     xlab = "Edad promedio", ylab = "Escolaridad promedio",
col="red", pch= 19,col.lab="blue",font.lab=3,font.main=
15)

text(edadYEscPorRegExp[,1],edadYEscPorRegExp[,2],
     labels = row.names(edadYEscPorRegExp) ,
     cex = .7,adj = c(0,0),pos = 1, font = 2)

legend(54.8,12.145,c("1:I", "2:II", "3:III", "4:IV",
            "5:V", "6:VI", "7:VII", "8:VIII",
            "9:IX", "10:X", "11:XI", "13:R.M",
            "14:XIV", "15:XV"),title = "Regiones",cex = .6,
```

```
        box.col = "azure3")
```