Práctica Calificada N.1

HUERTAS, ANTHONY[1,*]

[1] *Maestría en Estadística, Escuela de Posgrado, Pontificia Universidad Católica del Perú, Lima, Perú*
[*] *Cod: 20173728*

Compiled June 1, 2018

Profesor: Luis Benites

1. ¿CUÁL ES EL SUPUESTO DE ALEATORIEDAD CON EL QUE TRABAJA EL PAQUETE MICE?

Si bien el paquete **mice** nos provee de una técnica de imputación de datos multivariantes incompletos mediante ecuaciones encadenadas, el supuesto de aleatoriedad con el que trabaja dependerá de la escala de la variable que se imputa incorporando su relación con las demás variables habiéndose previamente especificado por defecto la cantidad de predictores a ser utilizados por cada variable incompleta. Además, el algoritmo MICE imputa por defecto columnas incompletas de datos de izquierda a derecha.

Cabe mencionar, que además de lo anteriormente dicho, **mice** requiere supuestos de modelado adicionales bajo datos **MNAR** que influyen en las imputaciones generadas.

El algoritmo MICE, particularmente, requiere de una especificación del método de imputación univariante separadamente para cada variable incompleta. Sin embargo, el paquete establece estas espeficicaciones por defecto pues reconoce variables numéricas, binarias, categóricas ordenadas y no ordenadas.

2. *IRIS2.CSV*, ANÁLISIS DE DATOS: DESCRIPTIVO, IMPUTACIONES, COMPARACIONES.

Dado que el archivo se encuentra en formato *.csv* se importarán los datos como se detalla en Listing 1, posteriormente se hace un resumen de los datos por variable correspondiente.

Listing 1. Importando datos.

```
datos_iris <- read.csv(file.choose())

# Se usa data.table para poder
# trabajar con sintaxis SQL
datos_iris <- data.table(datos_iris[,2:6])

# Se visualizan los 6 primeros datos
head(datos_iris)

# Resumen de datos por variable
summary(datos_iris)

# Tipos de variable
str(datos_iris)
```

```
   Sepal.Length Sepal.Width Petal.Length Petal.Width Species
1:          5.1         3.5           NA         0.2  setosa
2:          4.9          NA          1.4         0.2  setosa
3:          4.7         3.2          1.3         0.2  setosa
4:          4.6         3.1          1.5         0.2  setosa
5:          5.0         3.6          1.4         0.2  setosa
6:          5.4         3.9          1.7         0.4  setosa
```

(a) Primeros 6 datos

```
 Sepal.Length    Sepal.Width     Petal.Length    Petal.Width          Species
Min.   :4.300   Min.   :2.000   Min.   :1.100   Min.   :0.100   setosa    :50
1st Qu.:5.100   1st Qu.:2.800   1st Qu.:1.500   1st Qu.:0.300   versicolor:50
Median :5.800   Median :3.000   Median :4.200   Median :1.300   virginica :50
Mean   :5.843   Mean   :3.055   Mean   :3.624   Mean   :1.178
3rd Qu.:6.400   3rd Qu.:3.300   3rd Qu.:5.100   3rd Qu.:1.800
Max.   :7.900   Max.   :4.400   Max.   :6.900   Max.   :2.500
                NA's   :20      NA's   :30      NA's   :15
```

(b) Resumen por variable

```
Classes 'data.table' and 'data.frame':  150 obs. of  5 variables:
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 NA 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  NA 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 NA ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
 - attr(*, ".internal.selfref")=<externalptr>
```

(c) Tipos de variables

Fig. S1. Primera descripción de los datos (a),(b),(c).

En **Fig.S1(a)** podemos observar la presencia de valores faltantes representadas por "NA" que en efecto vienen resumidas en **Fig.S1(b)** por cada variable de los datos. Como observamos en **Fig.S1(b)** contamos con 3 tipos de especies (setosa,versicolor,virgínica) con 50 datos cada una. Tenemos una media global de 5.843 para la variable Sepal.Length; 3.055 para Sepal.Width; 3.624 para Petal.Length; 1.178 para Petal.Width. Sin embargo, podemos obtener las medias de cada variable por cada tipo de especie, pero no puede ser realizado de una forma simple a causa de presencia de NA's. Por lo que se optará por el siguiente mecanismo

Listing 2. Medias de cada variable sin contar los NA's.

```
d1 <- datos_iris[ Sepal.Length > 0 , mean(Sepal.Length), Species]
d2 <- datos_iris[ Sepal.Width  > 0 , mean(Sepal.Width), Species]
d3 <- datos_iris[ Petal.Length > 0, mean(Petal.Length), Species]
d4 <- datos_iris[ Petal.Width  > 0, mean(Petal.Width), Species]

d <- data.table(d1,d2[,2],d3[,2],d4[,2])
colnames(d) <- c("Species","Mean Sepal.Length","Mean Sepal.Width","Mean Petal.Length","Mean Petal
    .Width")
d
```

```
      Species Mean Sepal.Length Mean Sepal.Width Mean Petal.Length Mean Petal.Width
1:     setosa            5.006         3.434884          1.473333        0.2521739
2: versicolor            5.936         2.757778          4.278947        1.3191489
3:  virginica            6.588         2.983333          5.567568        2.0333333
```

Fig. S2. Medias de cada variable por Especie.

Hay que tomar en cuenta que las medidas resumidas anteriormente, son las extraídas de los datos habiéndose omitido las NA. En efecto, existen valores faltantes en las variables, y por tanto una pérdida de información. Realicemos entonces un primer análisis descriptivo de los datos omitiendo los valores NA.

A continuación, observemos la dispersión presente, gráficas de densidad y las correlaciones respectivas, entre cada una de las variables

Listing 3. Diagrama de dispersión global.

```
library(scatterplot3d)
library(GGally)
library(ggplot2)
ggpairs(datos_iris[,-5])+ theme_bw()
```

Como se puede obsevar en **Fig. S3**, las correlaciones entre las variables Sepal.Length-Petal.Length (0.875), Sepal.Length-Petal.Width (0.807), Petal.Length-Petal.Width (0.965) son altas. Sin embargo, con respecto a las otras variables, se presentan correlaciones relativamente bajas. Hay que tomar en cuenta que el análisis en este caso es global, por lo que, habiéndose 3 tipos de especies (Species), sería necesario evaluar las diferencias que se presentan respecto a dispersión (y correlación). Además, podemos observar los gráficos de densidad establecidos por cada variable continua de los datos de forma global.

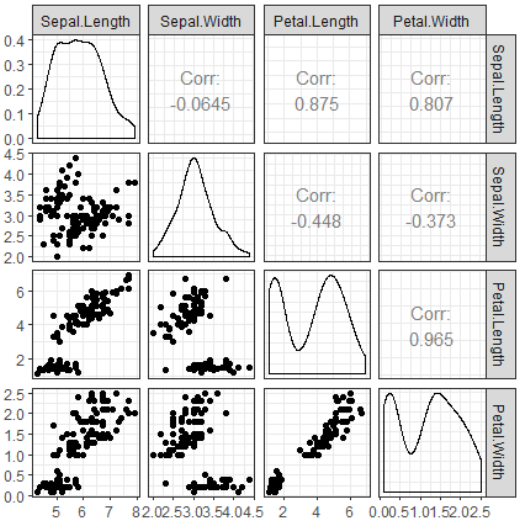


Fig. S3. Diagramas de dispersión y gráficos de densidad.

A continuación, se obtendrán los diagramas de dispersión, densidades, y gráficos de caja divididos por cada tipo de Especie

Listing 4. Diagrama de dispersión por cada variable.

```
p <- ggpairs(datos_iris, aes(color = Species))+ theme_bw()

for(i in 1:p$nrow) {
        for(j in 1:p$ncol){
                p[i,j] <- p[i,j] +
                  scale_fill_manual(values=c("#00AFBB", "#E7B800", "#FC4E07")) +
                  scale_color_manual(values=c("#00AFBB", "#E7B800", "#FC4E07"))
        }
}

p
```

Como logramos observar en **Fig.S4.**, las correlaciónes son totalmente distintas cuando evaluamos por la variable Especie (Species). Por ejemplo, como en el caso de Sepal.Length-Sepal.Width se presenta una correlación negativa y muy baja a nivel global (-0.0645); sin embargo se observan correlaciones relativamente altas cuando se establece la comparación con respecto a tipo de Especie, obteniendose correlaciones de 0.744(Setosa), 0.557 (versicolor), 0.596(virgínica). El mismo análisis es evaluado con respecto a los demás grupos

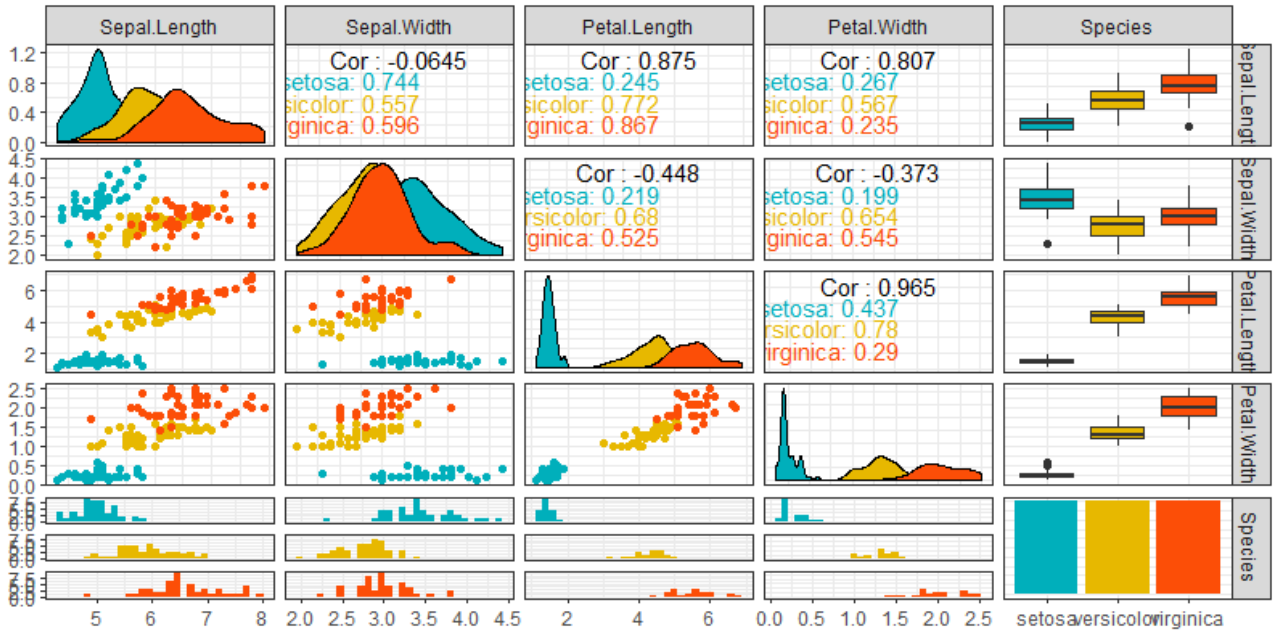


Fig. S4. Diagramas de dispersión por cada variable.

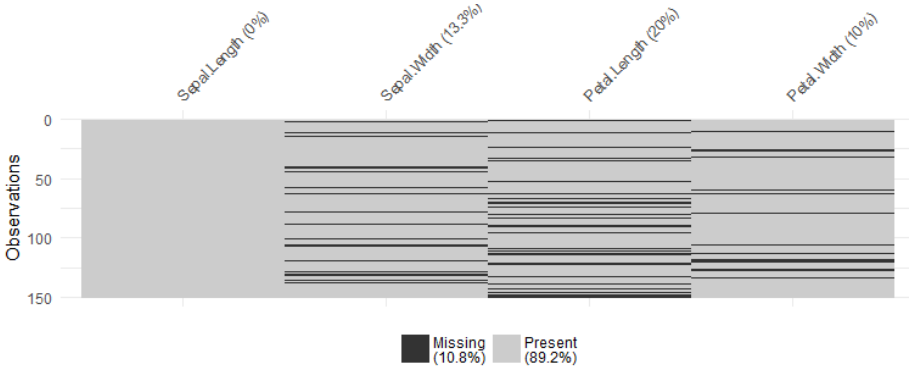
de dispersión, además observamos la formación de clusters representados por cada tipo de Especie. Con respecto a los gráficos de caja y densidades, podemos decir que es un resumen mucho más amplio y representativo que lo obtenido en **Fig.S2.** De igual forma podemos observar que nuestras densidad no tienen un alcance multivariado como se observaba en las densidades globales en **Fig.S3.**. Concluimos la importancia de la variable Species, y esto nos permitirá hacer uso esencial de esta en posteriores análisis.

Debido a que nuestros datos iniciales presentan valores "NA" como se observó en un inicio, entonces es necesario la evaluación respectiva de como se presentan en los datos (con respecto a ubicación) y los porcentajes de pérdida establecida por cada variable.

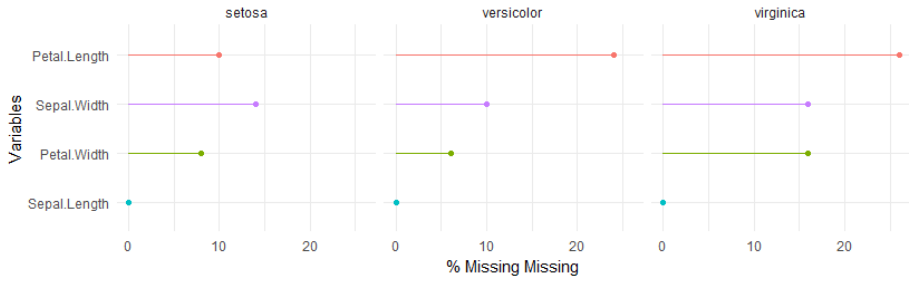
Listing 5. Datos faltantes.

```
vis_miss(datos_iris[,1:4])
gg_miss_var(datos_iris[,1:5],
            show_pct = TRUE,
            facet = Species)
md.pattern(datos_iris)
```

Observamos que no existe una pérdida de datos para la variable Sepal.Length, mientras que las demás variables si la presentan, generándose un mayor porcentaje de pérdida en la variable Petal.Length tanto globalmente como para las Especies versicolor y virgínica; para la especie setosa, la variable que genera mayor información pérdida es la de Sepal.Width.



(a) Ubicación de valores faltanes y porcentaje establecido de pérdida



(b) Gráfico de bastones porcentual de valores faltantes por Especies

Fig. S5. Datos faltantes por variable

Ahora se procederá a realizar el proceso de imputación usando el paquete mice.

Listing 6. Imputación con MICE.

```
tempData <- mice(datos_iris,
                 m=5,
                 maxit=50,
                 meth='pmm',
                 seed=500)
summary(tempData)

completedData <- complete(tempData,1)
str(completedData)
```

```
Multiply imputed data set
Call:
mice(data = datos_iris, m = 5, method = "pmm", maxit = 50,
    seed = 500)
Number of multiple imputations:  5
Missing cells per column:
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
           0           20           30           15           0
Imputation methods:
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
       "pmm"        "pmm"        "pmm"        "pmm"        "pmm"
VisitSequence:
 Sepal.Width Petal.Length  Petal.Width
           2            3            4
PredictorMatrix:
             Sepal.Length Sepal.Width Petal.Length Petal.Width Species
Sepal.Length            0           0            0           0       0
Sepal.Width             1           0            1           1       1
Petal.Length            1           1            0           1       1
Petal.Width             1           1            1           0       1
Species                 0           0            0           0       0
Random generator seed value:  500
```

(a) Resumen

```
'data.frame':   150 obs. of  6 variables:
 $ X           : int  1 2 3 4 5 6 7 8 9 10 ...
 $ Sepal.Length: num  5.1 4.9 4.7 4.6 5 5.4 4.6 5 4.4 4.9 ...
 $ Sepal.Width : num  3.5 3.1 3.2 3.1 3.6 3.9 3.4 3.4 2.9 3.1 ...
 $ Petal.Length: num  1.6 1.4 1.3 1.5 1.4 1.7 1.4 1.5 1.4 1.5 ...
 $ Petal.Width : num  0.2 0.2 0.2 0.2 0.2 0.4 0.3 0.2 0.2 0.3 ...
 $ Species     : Factor w/ 3 levels "setosa","versicolor",..: 1 1 1 1 1 1 1 1 1 1 ...
  ..- attr(*, "contrasts")= num [1:3, 1:2] 0 1 0 0 0 1
  .. ..- attr(*, "dimnames")=List of 2
  .. .. ..$ : chr  "setosa" "versicolor" "virginica"
  .. .. ..$ : chr  "2" "3"
```

(b) Tipo de variable

Fig. S6. Descripción de datos con valores imputados.

En efecto, el mecanismo de imputación ha generado que los valores generados sean reemplazados por valores adecuados. Verificándose a continuación que se presenta ahora datos sin valores faltantes

Listing 7. Verificación de valores faltantes.

```
sapply(completedData, function(x) sum(is.na(x)))
```

```
Sepal.Length  Sepal.Width Petal.Length  Petal.Width      Species
           0           0            0           0           0
```

Fig. S7. Conteo de valores faltantes por variable.

A continuación, se obtendrán los diagramas de dispersión de los datos con valores imputados asignados

Listing 8. Diagrama de dispersión global con valores imputados.

```
ggpairs(completedData[,-5])+ theme_bw()
```

Como se puede obsevar en **Fig.S8**, las correlaciones globales respecto a estos nuevos datos con valores imputados son similares a las obtenidas por los datos iniciales en **Fig.S3**.

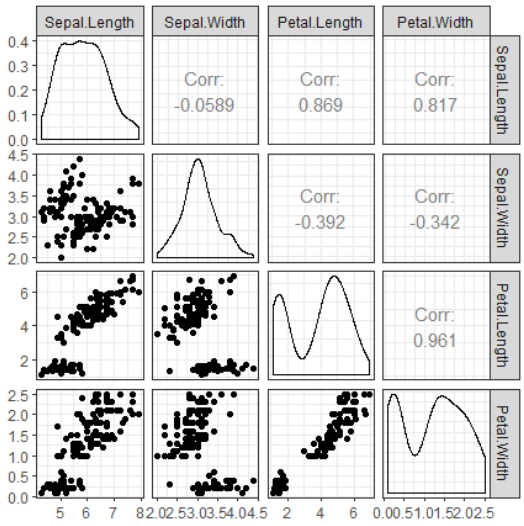


Fig. S8. Datos imputados: Diagramas de dispersión y gráficos de densidad.

A continuación, se obtendrán los diagramas de dispersión, densidades, y gráficos de caja divididos por cada tipo de Especie, de los datos con valores imputados.

Listing 9. Diagrama de dispersión por cada variable con valores imputados.

```
p <- ggpairs(completedData, aes(color = Species))+ theme_bw()

for(i in 1:p$nrow) {
        for(j in 1:p$ncol){
                p[i,j] <- p[i,j] +
                scale_fill_manual(values=c("#00AFBB", "#E7B800", "#FC4E07")) +
                scale_color_manual(values=c("#00AFBB", "#E7B800", "#FC4E07"))
        }
}

p
```

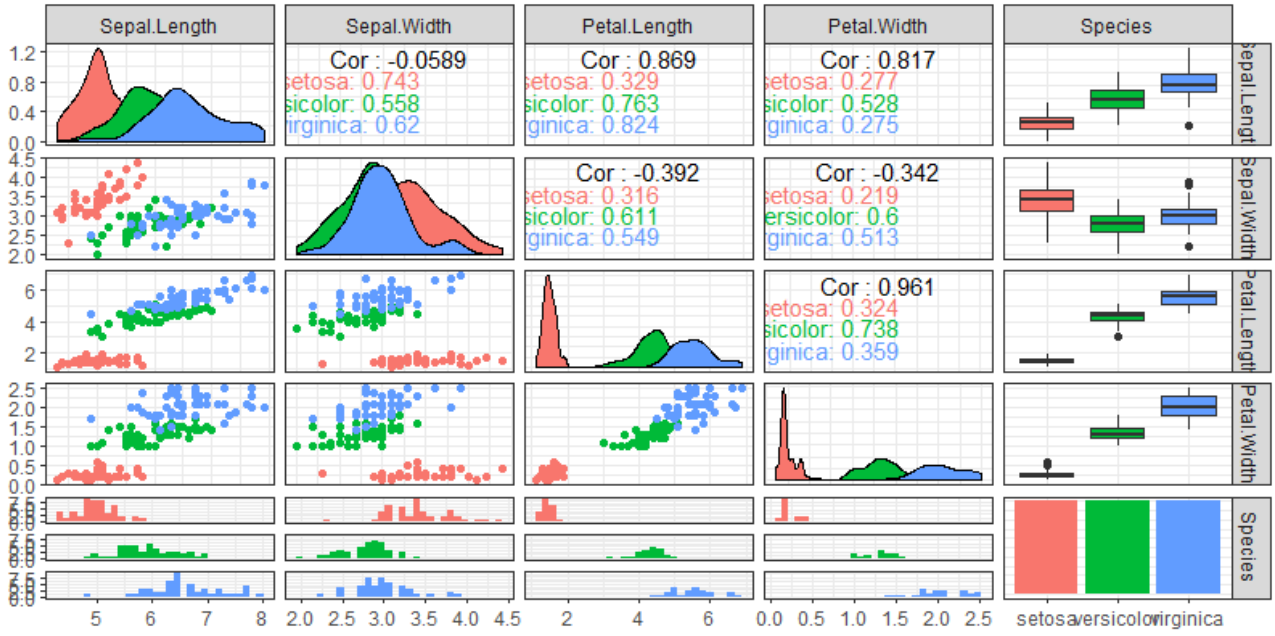


Fig. S9. Datos imputados: Diagramas de dispersión por cada variable.

Como se observa, valores nuevos han sido asignados a los valores faltantes mediante el mecanismo de imputación propuesto por mice, generando una nueva base de datos con una similitud general con respecto a los datos iniciales.La imputación realizada mantiene la estabilidad de los resultados observados gráficamente,

Hay que tener en claro que existe una base de datos original en R, sin valores faltantes, de donde proviene nuestra data inicial con valores faltantes en este problema. Debido a la imputación realizada bajo los datos con valores faltantes, analicemos el RMSE normalizado en comparación con los datos originales.

Listing 10. RMSE.

```
Rmse(completedData[,1:4],as.data.frame(datos_iris)[,1:4], iris[,1:4], norm = TRUE)
```

$$nRMSE = 0.2138646$$

siendo un valor bajo, indicando el valor promedio de los errores al cuadrado entre los datos reales y los datos con imputaciones de valores. Podemos decir que la imputación es aceptable.

3. INCORPORACIÓN DE DATOS PERDIDOS A UN CONJUNTO DE DATOS

Se hizo uso de un conjunto de datos referente a pacientes que han sido evaluados para diabetes.

Listing 11. Importando datos.

```
diabetes <- read.csv(file.choose(),
                     header=TRUE,
                     sep=",")
dim(diabetes)
head(diabetes)
str(diabetes)

# Duplicando datos
original <- diabetes
```

Tenemos un conjunto de datos de 115 observaciones, 3 variables de tipo numérica y 1 de tipo factor con 3 niveles.

```
  glucose insulin sspg  class
1      97     289  117 normal
2     105     319  143 normal
3      90     356  199 normal
4      90     323  240 normal
5      86     381  157 normal
6     100     350  221 normal
```

(a) Primeros 6 datos

```
'data.frame':   115 obs. of  4 variables:
 $ glucose: int  97 105 90 90 86 100 85 97 97 91 ...
 $ insulin: int  289 319 356 323 381 350 301 379 296 353 ...
 $ sspg   : int  117 143 199 240 157 221 186 142 131 221 ...
 $ class  : Factor w/ 3 levels "chemical","normal",..: 2 2 2 2 2 2 2 2 2 2 ...
```

(b) Tipos de variable

Fig. S10. Primera descripción de los datos (a),(b).

Vemos que estamos tratando con un conjunto de datos sin valores faltantes.

Listing 12. Importando datos.

```
sapply(diabetes, function(x) sum(is.na(x)))
```

```
glucose insulin    sspg   class
      0       0       0       0
```

Fig. S11. Cantidad de valores faltantes por variable, de la base original.

Como logramos observar, no existen valores faltantes en cada variable por lo que iniciaremos con lo propuesto por el problema, lo cual es la incorporación de valores faltantes, verificando posteriormente que el mecanismo se realizó correctamente.

Listing 13. Incorporación de valores faltantes.

```
set.seed(10)
diabetes[sample(1:nrow(diabetes), 20), "glucose"] <- NA
diabetes[sample(1:nrow(diabetes), 20), "insulin"] <- NA
diabetes[sample(1:nrow(diabetes), 20), "sspg"] <- NA
diabetes[sample(1:nrow(diabetes), 5), "class"] <- NA

#Confirme la presencia de errores en el conjunto de datos.

sapply(diabetes, function(x) sum(is.na(x)))
```

```
glucose insulin    sspg   class
     20      20      20       5
```

Fig. S12. Cantidad de valores faltantes por variable, de la base rediseñada con valores faltantes.

Se imputarán datos mediante las técnicas mice e ImputeR, permitiendo generar datos sin valores faltantes, con lo cual evaluaremos posteriormente con nuestros datos originales.

Listing 14. Imputación.

```
##------ Mice
library(mice)
init = mice(diabetes, maxit=0)
meth = init$method
predM = init$predictorMatrix

meth[c("glucose")]="norm"
meth[c("insulin")]="norm"
meth[c("sspg")]="norm"
meth[c("class")]="polyreg"

set.seed(103)
imputed = mice(diabetes, method=meth, predictorMatrix=predM, m=15)
imputed <- complete(imputed)

##------ImputeR
imputed.R <- data.table(impute(diabetes[,1:3], lmFun = "lassoR")$imp)
```

Ahora se evaluará la exactitud provista por las tres técnicas, considerando que la última solo ha imputado bajo las variables numéricas.

Listing 15. Exactitud.

```
M <- c(mean(original$glucose[is.na(diabetes$glucose)]),
       mean(imputed$glucose[is.na(diabetes$glucose)]),
       mean(imputed.R$glucose[is.na(diabetes$glucose)]))
M <- rbind(M,c(mean(original$insulin[is.na(diabetes$insulin)]),
               mean(imputed$insulin[is.na(diabetes$insulin)]),
               mean(imputed.R$insulin[is.na(diabetes$insulin)])))
M <- rbind(M,c(mean(original$sspg[is.na(diabetes$sspg)]),
               mean(imputed$sspg[is.na(diabetes$sspg)]),
               mean(imputed.R$sspg[is.na(diabetes$sspg)])))
colnames(M) <- c("original","mice","imputeR")
rownames(M) <- c("Mean.glucose","Mean.insulin","Mean.sspg")
M
```

```
              original      mice   imputeR
Mean.glucose     93.70  98.17007  97.28515
Mean.insulin    448.20 463.93231 488.45160
Mean.sspg       136.95 189.36021 187.23047
```

Fig. S13. Exactitud de medias.

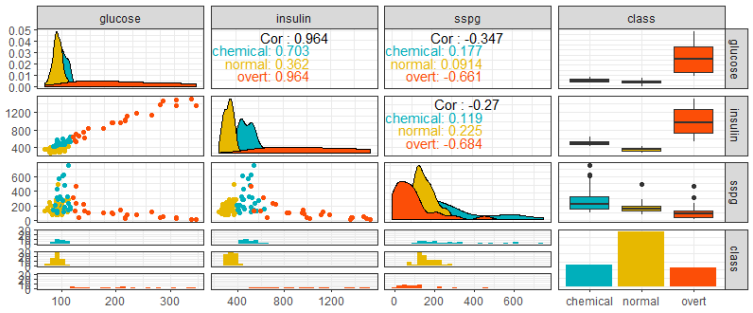
Como notamos, las técnicas mice e imputeR tienen una gran exactitud respecto a la media de los valores que habían sido asignados perdidos en los datos originales.

Diseñemos gráficos de dispersión y demás respecto de las variables de los datos con valores imputados.

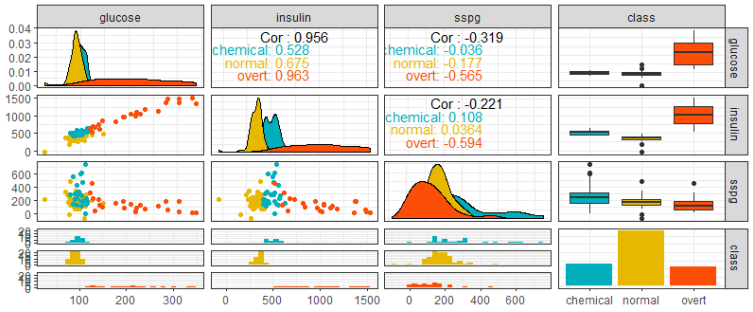
Listing 16. Diagrama de dispersión por cada variable con valores imputados.

```
p <- ggpairs(original, aes(color = class))+ theme_bw()
for(i in 1:p$nrow) {
        for(j in 1:p$ncol){
                p[i,j] <- p[i,j] +
                scale_fill_manual(values=c("#00AFBB", "#E7B800", "#FC4E07")) +
                scale_color_manual(values=c("#00AFBB", "#E7B800", "#FC4E07"))
        }
}
p

p2 <- ggpairs(imputed, aes(color = class))+ theme_bw()
for(i in 1:p$nrow) {
        for(j in 1:p$ncol){
                p[i,j] <- p[i,j] +
                scale_fill_manual(values=c("#00AFBB", "#E7B800", "#FC4E07")) +
                scale_color_manual(values=c("#00AFBB", "#E7B800", "#FC4E07"))
        }
}
p2
```



(a) Datos originales: Diagramas de dispersión por cada variable y clase.



(b) Datos imputados con mice: Diagramas de dispersión por cada variable y clase.

Fig. S14. Gráficos de dispersión, densidad y diagramas de caja.

Observamos que de las imputaciones realizadas se han generado valores tales que a nivel global se establecen correlaciones casi iguales. Sin embargo, a nivel de clase, las correlaciones si se ven afectadas en algunos casos. La tendencia de correlación se mantiene al igual que los cluster pero al parecer no es tan correcta si observamos las densidades establecidas en ambos gráficos, pues observamos que varían considerablemente. Un buen segundo análisis podría ser el aumentar el número de iteraciones en el mecanismo mice. En general, gráficamente los datos son completados de forma aceptable.

4. *MSLEEP*, ANÁLISIS DE DATOS: DESCRIPTIVO, IMPUTACIONES, COMPARACIONES.

Primero se importan los datos, proveniente del paquete ggplot2

Listing 17. Importando datos.

```
data("msleep", package = "ggplot2")
```

Se análizará si los datos en estudio contiene valores faltantes en su estructura, para ello realizamos lo siguiente

Listing 18. Datos faltantes.

```
aggr_plot<-aggr(msleep, col=c('navyblue','red'),
                numbers=TRUE, sortVars=TRUE,
                labels=names(msleep), cex.axis=.7,
                gap=3,
                ylab=c("Histogram of missing data",
                       "Pattern"))
```

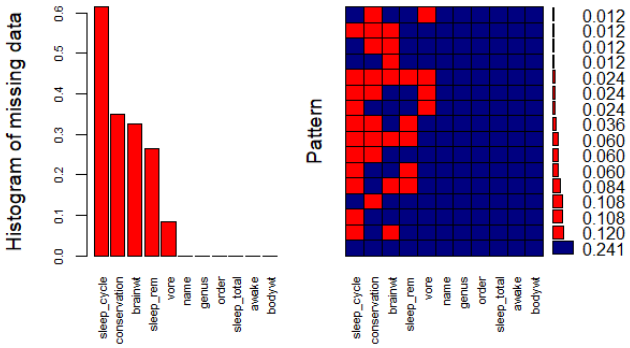


Fig. S15. Datos faltantes por variable.

Como podemos observar, si existen valores faltantes en los datos, indicando a su vez que se presenta un porcentaje muy alto de estos (aproximadamente 60%) en la variable sleep_cycle.

Veamos un resumen descriptivo de los datos antes de iniciar el proceso de imputación.

Listing 19. Estructura de los datos.

```
msleep <- data.table(msleep)

# Cambiando de tipo ch a tipo factor
datos <- msleep %>% mutate(name = as.factor(name)) %>% mutate(genus = as.factor(genus)) %>%
         mutate(vore = as.factor(vore)) %>% mutate(order = as.factor(order)) %>%
         mutate(conservation = as.factor(conservation))
str(datos)
summary(datos)
```

```
classes 'tbl_df', 'tbl' and 'data.frame':       83 obs. of  11 variables:
 $ name        : Factor w/ 83 levels "African elephant",..: 12 57 52 36 17 77 55 81 21 6
7 ...
 $ genus       : Factor w/ 77 levels "Acinonyx","Aotus",..: 1 2 3 4 5 6 7 8 9 10 ...
 $ vore        : Factor w/ 4 levels "carni","herbi",..: 1 4 2 4 2 2 1 NA 1 2 ...
 $ order       : Factor w/ 19 levels "Afrosoricida",..: 3 15 17 19 2 14 3 17 3 2 ...
 $ conservation: Factor w/ 6 levels "cd","domesticated",..: 4 NA 5 4 2 NA 6 NA 2 4 ...
 $ sleep_total : num  12.1 17 14.4 14.9 4 14.4 8.7 7 10.1 3 ...
 $ sleep_rem   : num  NA 1.8 2.4 2.3 0.7 2.2 1.4 NA 2.9 NA ...
 $ sleep_cycle : num  NA NA NA NA 0.133 0.667 ...
 $ awake       : num  11.9 7 9.6 9.1 20 9.6 15.3 17 13.9 21 ...
 $ brainwt     : num  NA 0.0155 NA 0.00029 0.423 NA NA NA 0.07 0.0982 ...
 $ bodywt      : num  50 0.48 1.35 0.019 600 ...
```

(a) Tipos de variable

```
                      name              genus         vore            order
 African elephant         : 1   Panthera    : 3   carni  :19   Rodentia    :22
 African giant pouched rat: 1   Spermophilus: 3   herbi  :32   Carnivora   :12
 African striped mouse    : 1   Equus       : 2   insecti: 5   Primates    :12
 Arctic fox               : 1   Vulpes      : 2   omni   :20   Artiodactyla: 6
 Arctic ground squirrel   : 1   Acinonyx    : 1   NA's   : 7   Soricomorpha: 5
 Asian elephant           : 1   Aotus       : 1                Cetacea     : 3
 (Other)                  :77   (Other)     :71                (Other)     :23
       conservation  sleep_total      sleep_rem      sleep_cycle         awake
 cd          : 2    Min.   : 1.90   Min.   :0.100   Min.   :0.1167   Min.   : 4.10
 domesticated:10    1st Qu.: 7.85   1st Qu.:0.900   1st Qu.:0.1833   1st Qu.:10.25
 en          : 4    Median :10.10   Median :1.500   Median :0.3333   Median :13.90
 lc          :27    Mean   :10.43   Mean   :1.875   Mean   :0.4396   Mean   :13.57
 nt          : 4    3rd Qu.:13.75   3rd Qu.:2.400   3rd Qu.:0.5792   3rd Qu.:16.15
 vu          : 7    Max.   :19.90   Max.   :6.600   Max.   :1.5000   Max.   :22.10
 NA's        :29                    NA's   :22      NA's   :51
    brainwt             bodywt
 Min.   :0.00014   Min.   :   0.005
 1st Qu.:0.00290   1st Qu.:   0.174
 Median :0.01240   Median :   1.670
 Mean   :0.28158   Mean   : 166.136
 3rd Qu.:0.12550   3rd Qu.:  41.750
 Max.   :5.71200   Max.   :6654.000
 NA's   :27
```

(b) Resumen

Fig. S16. Resumen de datos

Ahora diseñaremos un proceso de imputación y evaluaremos el modelo de regresión de ciertas variables haciendo uso de la nueva base de datos

Listing 20. Imputación y evaluación en modelo de regresión

```
set.seed(103)
imputed = mice(datos, seed=103)

fit <- with(imputed, lm(sleep_total ~ awake + factor(vore)))
print(pool(fit))
round(summary(pool(fit)), 2)
```

```
                      est se       t    df Pr(>|t|) lo 95 hi 95 nmis  fmi lambda
(Intercept)         24.00  0 7778.85 75.88     0.00 23.99 24.01   NA 0.03   0.00
awake               -1.00  0 -5271.48 75.79     0.00 -1.00 -1.00    0 0.03   0.00
factor(vore)herbi   -0.01  0    -2.68 76.01     0.01 -0.01  0.00   NA 0.03   0.00
factor(vore)insecti  0.00  0    -1.05 74.37     0.30 -0.01  0.00   NA 0.04   0.02
factor(vore)omni    -0.01  0    -2.18 76.02     0.03 -0.01  0.00   NA 0.03   0.00
```

Fig. S17. Coeficientes de regresión.

Tenemos como resultados, estimaciones de parámetros de regresión lineal entre ciertas variables, con grados de significancia aceptables.

5. *HTTPS://GOO.GL/IR3DIF*, ANÁLISIS DE DATOS: DESCRIPTIVO, IMPUTACIONES, COMPARACIONES.

Los datos corresponden a los indicadores sociales de las Naciones Unidas.

Listing 21. Importando datos.

```
dat <- read.csv(url("https://goo.gl/
                iR3diF"),
                header=TRUE,
                sep=" ")
dat <- data.table(dat)
str(dat)
summary(dat)
```

Como se observa, solo se cuentan con variables de tipo entero (int) y numérico (num), y una variable de tipo (factor)

```
'data.frame':   207 obs. of  13 variables:
$ region                : Factor w/ 5 levels "Africa","America",..: 3 4 1 3 4 1 2 2 4 5
...
$ tfr                   : num  6.9 2.6 3.81 NA NA 6.69 NA 2.62 1.7 1.89 ...
$ contraception         : int  NA NA 52 NA NA NA 53 NA 22 76 ...
$ educationMale         : num  NA NA 11.1 NA NA NA NA NA NA 16.3 ...
$ educationFemale       : num  NA NA 9.9 NA NA NA NA NA NA 16.1 ...
$ lifeMale              : num  45 68 67.5 68 NA 44.9 NA 69.6 67.2 75.4 ...
$ lifeFemale            : num  46 74 70.3 73 NA 48.1 NA 76.8 74 81.2 ...
$ infantMortality       : int  154 32 44 11 NA 124 24 22 25 6 ...
$ GDPperCapita          : int  2848 863 1531 NA NA 355 6966 8055 354 20046 ...
$ economicActivityMale  : num  87.5 NA 76.4 58.8 NA NA 74.4 76.2 65 74 ...
$ economicActivityFemale: num  7.2 NA 7.8 42.4 NA NA 56.2 41.3 52 53.8 ...
$ illiteracyMale        : num  52.8 NA 26.1 0.264 NA NA NA 3.8 0.3 NA ...
$ illiteracyFemale      : num  85 NA 51 0.36 NA NA NA 3.8 0.5 NA ...
```

Fig. S18. Estructura de los datos.

Listing 22. Cantidad de valores faltantes.

```
sapply(dat, function(x) sum(is.na(x)))
```

Se observan altas cantidades de valores “NA” para las covariables. Será importante identificar los porcentajes de pérdida respecto a cada variable.

```
            region                tfr        contraception
                 0                 10                 63
    educationMale      educationFemale          lifeMale
               131                131                 11
        lifeFemale       infantMortality       GDPperCapita
                11                  6                 10
economicActivityMale economicActivityFemale    illiteracyMale
                42                 42                 47
  illiteracyFemale
                47
```

Fig. S19. Valores faltantes por variable.

Listing 23. Datos faltantes.

```
aggr_plot<-aggr(dat, col=c('navyblue','red'),
                numbers=TRUE, sortVars=TRUE,
                labels=names(dat), cex.axis=.7,gap=3,
                ylab=c("Histogram of missing data","Pattern"))
```

Los porcentajes de valores perdidos son de aproximadamente del 60% en las primeras variables, por lo que podemos indicar que hay una pérdida importante de datos.

```
Variables sorted by number of missings:
              Variable      Count
         educationMale 0.63285024
       educationFemale 0.63285024
         contraception 0.30434783
        illiteracyMale 0.22705314
      illiteracyFemale 0.22705314
  economicActivityMale 0.20289855
economicActivityFemale 0.20289855
           lifeMale 0.05314010
         lifeFemale 0.05314010
                tfr 0.04830918
          GDPperCapita 0.04830918
     infantMortality 0.02898551
              region 0.00000000
```

Fig. S20. Porcentaje de Valores faltantes por variable.

A continuación, se visualizan los diagramas de dispersión entre cada variable para evaluar gráficamente posibles correlaciones presentes. Claramente se analizará gráficamente omitiendo en primera instancia los valores "NA".Debido a la alta dimensionalidad de variables en los datos, se hará una gráfica de dispersión estándar en la que se observará en forma global.

Listing 24. Gráficos de dispersión

```
plot(dat)
```

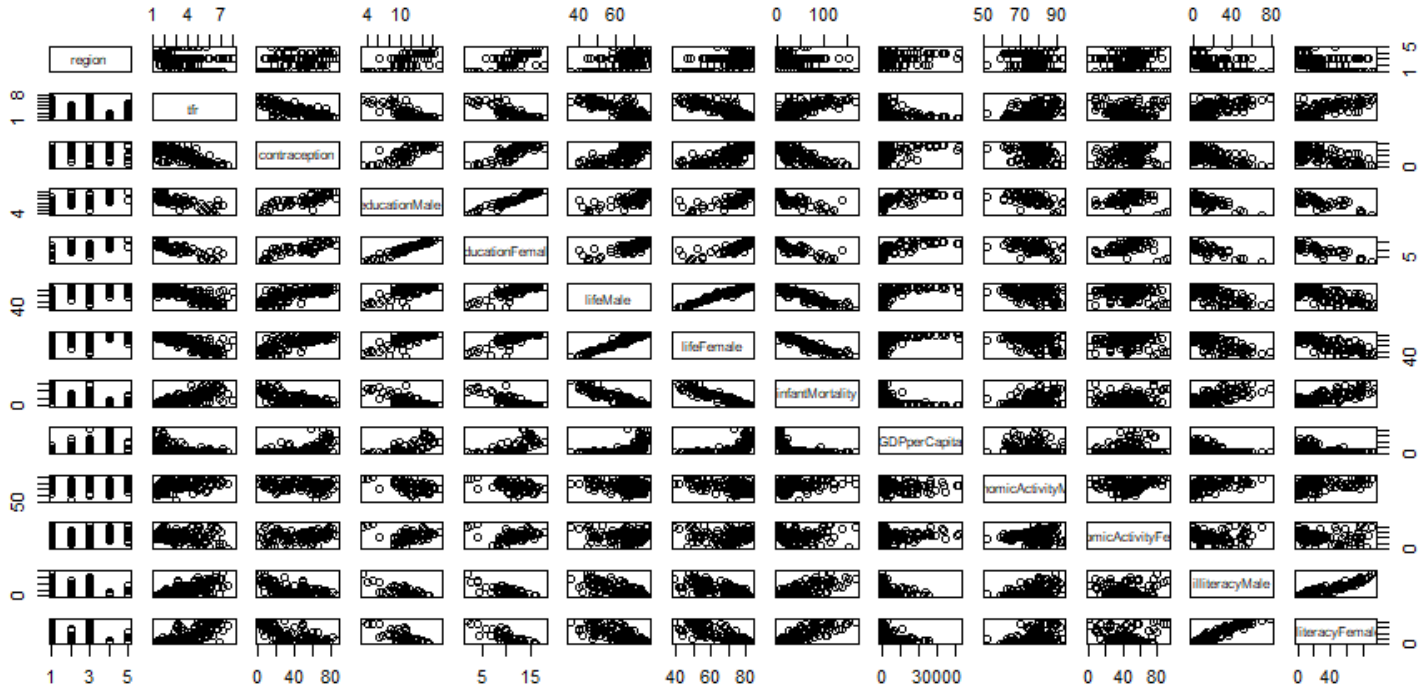


Fig. S21. Diagrama de dispersión.

Se pueden observar claras correlaciones positivas y negativas entre ciertas variables.

Se realizará el proceso de imputación mediante el paquete mice e ImputeR con el objeto podamos comparar ambas técnicas. Ambas serán comparadas con las medias obtenidas respecto a cada variable entera y numérica. Sin embargo, con la primera técnica se hará uso de la variable categórica en el mecanismo mientras que en la segunda técnica no.

Listing 25. Imputación.

```
##------ Mice
library(mice)
init = mice(dat, maxit=0)
meth = init$method
predM = init$predictorMatrix

meth[c("illiteracyFemale")]="norm"
meth[c("illiteracyMale")]="norm"
meth[c("economicActivityFemale")]="norm"
meth[c("economicActivityMale")]="norm"
meth[c("GDPperCapita")]="pmm"
meth[c("infantMortality")]="pmm"
meth[c("lifeFemale")]="norm"
meth[c("lifeMale")]="norm"
meth[c("educationFemale")]="norm"
meth[c("educationMale")]="norm"
meth[c("contraception")]="pmm"
meth[c("tfr")]="norm"
meth[c("region")]="polyreg"

set.seed(13)
imputed = mice(dat, method=meth, predictorMatrix=predM, m=20)
imputed <- complete(imputed)


```

```r
##------ImputeR
library("imputeR")
imputed.R <- data.table(impute(dat[,2:13], lmFun = "lassoR")$imp)
```

Ahora se evaluará la media de valores que resultan de la imputación sobre los valores faltantes, como proceso de comparación entre ambos modelos.

Listing 26. Exactitud.

```r
M <- c(mean(imputed$tfr[is.na(dat$tfr)]),
       mean(imputed.R$tfr[is.na(dat$tfr)]))
M <- rbind(M,c(mean(imputed$contraception[is.na(dat$contraception)]),
               mean(imputed.R$contraception[is.na(dat$contraception)])))
M <- rbind(M,c(mean(imputed$educationMale[is.na(dat$educationMale)]),
               mean(imputed.R$educationMale[is.na(dat$educationMale)])))
M <- rbind(M,c(mean(imputed$educationFemale[is.na(dat$educationFemale)]),
               mean(imputed.R$educationFemale[is.na(dat$educationFemale)])))
M <- rbind(M,c(mean(imputed$lifeMale[is.na(dat$lifeMale)]),
               mean(imputed.R$lifeMale[is.na(dat$lifeMale)])))
M <- rbind(M,c(mean(imputed$lifeFemale[is.na(dat$lifeFemale)]),
               mean(imputed.R$lifeFemale[is.na(dat$lifeFemale)])))
M <- rbind(M,c(mean(imputed$infantMortality[is.na(dat$infantMortality)]),
               mean(imputed.R$infantMortality[is.na(dat$infantMortality)])))
M <- rbind(M,c(mean(imputed$GDPperCapita[is.na(dat$GDPperCapita)]),
               mean(imputed.R$GDPperCapita[is.na(dat$GDPperCapita)])))
M <- rbind(M,c(mean(imputed$economicActivityMale[is.na(dat$economicActivityMale)]),
               mean(imputed.R$economicActivityMale[is.na(dat$economicActivityMale)])))
M <- rbind(M,c(mean(imputed$economicActivityFemale[is.na(dat$economicActivityFemale)]),
               mean(imputed.R$economicActivityFemale[is.na(dat$economicActivityFemale)])))
M <- rbind(M,c(mean(imputed$illiteracyMale[is.na(dat$illiteracyMale)]),
               mean(imputed.R$illiteracyMale[is.na(dat$illiteracyMale)])))
M <- rbind(M,c(mean(imputed$illiteracyFemale[is.na(dat$illiteracyFemale)]),
               mean(imputed.R$illiteracyFemale[is.na(dat$illiteracyFemale)])))

colnames(M) <- c("mice","imputeR")
dim(M)
rownames(M) <- c("Mean.tfr","Mean.contraception","Mean.educationMale",
                 "Mean.educationFemale","Mean.lifeMale","Mean.lifeFemale",
                 "Mean.infantMortality","Mean.GDPperCapita","Mean.economicActivityMale",
                 "Mean.economicActivityFemale","Mean.illiteracyMale","Mean.illiteracyFemale")
M
```

```
                              mice     imputeR
Mean.tfr                  2.846096    2.969873
Mean.contraception       47.126984   44.413092
Mean.educationMale       11.186969   10.707113
Mean.educationFemale     10.749257   10.271931
Mean.lifeMale            66.588335   65.043530
Mean.lifeFemale          71.295601   70.239063
Mean.infantMortality     28.166667   34.289497
Mean.GDPperCapita      7431.300000 6320.515410
Mean.economicActivityMale 78.901562   79.105185
Mean.economicActivityFemale 48.702585   45.647324
Mean.illiteracyMale       8.531209   15.716652
Mean.illiteracyFemale    16.704832   22.871864
```

Fig. S22. Medias de valores imputados.

Como notamos, las técnicas mice e imputeR generan una media de valores imputados semejantes en todas las variables excepto en la media con respecto a la variable GDPperCapita que existe una diferencia significativa, al igual que para la variable illiteracyMale, en donde claramente el mecanismo por imputeR genera una media de casi el doble que el mecanismo mice en los valores faltantes. En estos último casos, sería ideal la estandarización para evitar tratar y reanalizar el mecanismo.