Análisis de encuestas de hogares con R

Modulo 9: Métodos de imputación

CEPAL - Unidad de Estadísticas Sociales

Tabla de contenidos I

Introducción

Introducción valores perdidos

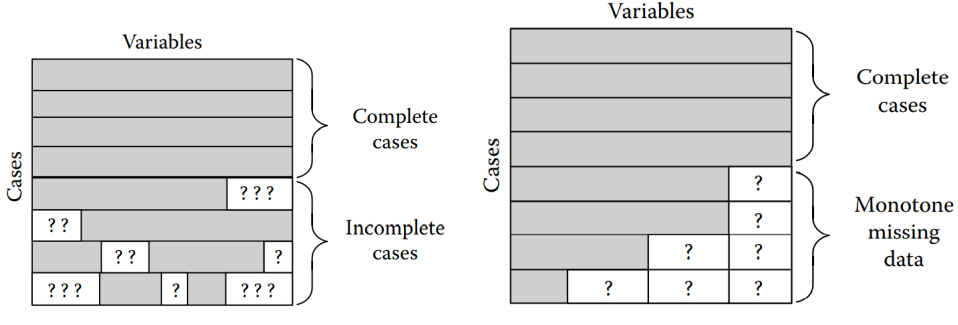
- Sea $X_{n\times p} = x_{ij}$ una matriz completa (sin valores perdidos), de tal forma que X_{ij} es el valor de la variable j, $j = 1, \dots, p$ en el caso i, $i = 1, \dots, n$.
- Sea $M_{n\times p} = m_{ij}$, donde $m_{ij} = 1$ si x_{ij} es un dato perdido y $m_{ij} = 0$ si x_{ij} está presente.
- Note que la matriz M describe el patrón de missing, y su media marginal de columna, puede ser interpretada como la probabilidad de que x_{ij} sea missing.

Introducción valores perdidos

- La matriz $M_{n\times p}$ presenta un comportamiento completamente al azar (MCAR): si la probabilidad de respuesta es independiente de las variables observadas y de las no observadas completamente. El mecanismo de pérdida es ignorable tanto para inferencias basadas en muestreo como en máxima verosimilitud.
- Los valores de la matriz $M_{n\times p}$ son al azar (MAR): si la probabilidad de respuesta es independiente de las variables no observadas completamente y no de las observadas. El mecanismo de pérdida es ignorable para inferencias basadas en máxima verosimilitud.
- Los datos no están perdidos al azar (MNAR): si la probabilidad de respuesta no es independiente de las variables no observadas completamente y posiblemente, también, de las observadas El mecanismo de pérdida es no ignorable.

Introducción valores perdidos

En las dos figuras siguientes, se ilustran los casos de observaciones perdidas de manera aleatoria y con un patrón identificado:



Lectura de la base

De la base de datos cargada se filtran encuestados mayores a 15 años y se calcula la proporción de la población desempleada, inactiva y empleada antes de generar los valores faltantes

```
encuesta <- readRDS("../Data/encuesta.rds") %>%
  filter(Age >= 15)
(tab_antes <- prop.table(table(encuesta$Employment)))
```

Unemployed	Inactive	Employed
0.041	0.3736	0.5854

```
(med_antes <- mean(encuesta$Income, na.rm = TRUE))
```

```
[1] 604.2
```

Creando valores perdidos

Se genera un 20% de valores faltantes siguiendo un esquema MCAR como sigue:

```
set.seed(1234)
encuesta_MCAR <-  sample_frac(encuesta, 0.8 )
dat_plot <- bind_rows(
  list(encuesta_MCAR = encuesta_MCAR,
       encuesta = encuesta), .id = "Caso"  )
```

Creando valores perdidos

```
p1 <- ggplot(dat_plot, aes(x=Zone, y = Income)) +
  geom_boxplot() + facet_grid(.~Caso) + theme_bw()+
  geom_hline(yintercept = mean(encuesta$Income),
             col = "red")

p2 <- ggplot(dat_plot, aes(x=Sex, y = Income)) +
  geom_boxplot() + facet_grid(.~Caso) +theme_bw()+
  geom_hline(yintercept = mean(encuesta$Income),
             col = "red")
library(patchwork)
p1|p2
```

Creando valores perdidos

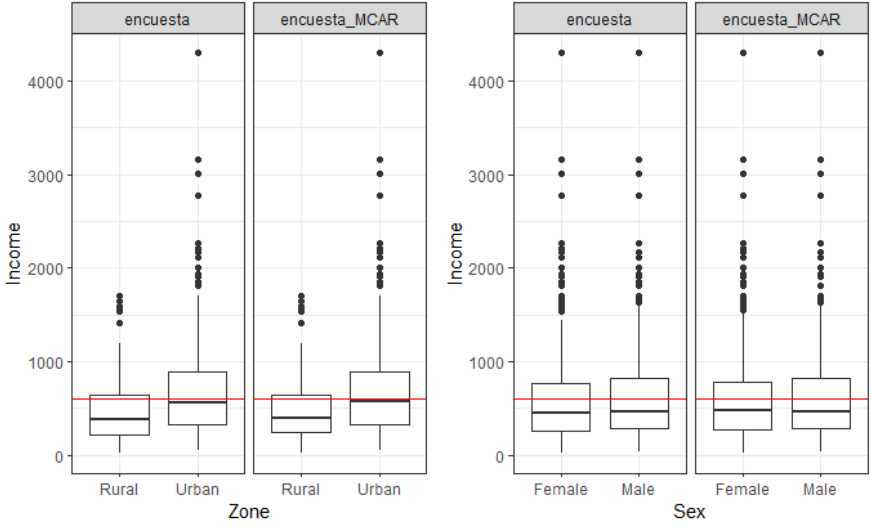


Figura 1: Valores perdidos con el esquema MCAR

Creando valores perdidos

```
p1 <- ggplot(dat_plot, aes(x = Income, fill = Caso)) +
  geom_density(alpha = 0.3) + theme_bw() +
  theme(legend.position = "bottom") +
  geom_vline(xintercept = mean(encuesta$Income),
             col = "red")

p2 <- ggplot(dat_plot, aes(x = Income, fill = Caso)) +
  geom_density(alpha = 0.3) + facet_grid(.~Sex) +
  theme_bw()+
  geom_vline(xintercept = mean(encuesta$Income),
             col = "red") +
  theme(legend.position = "none")
(p1/p2)
```

Creando valores perdidos

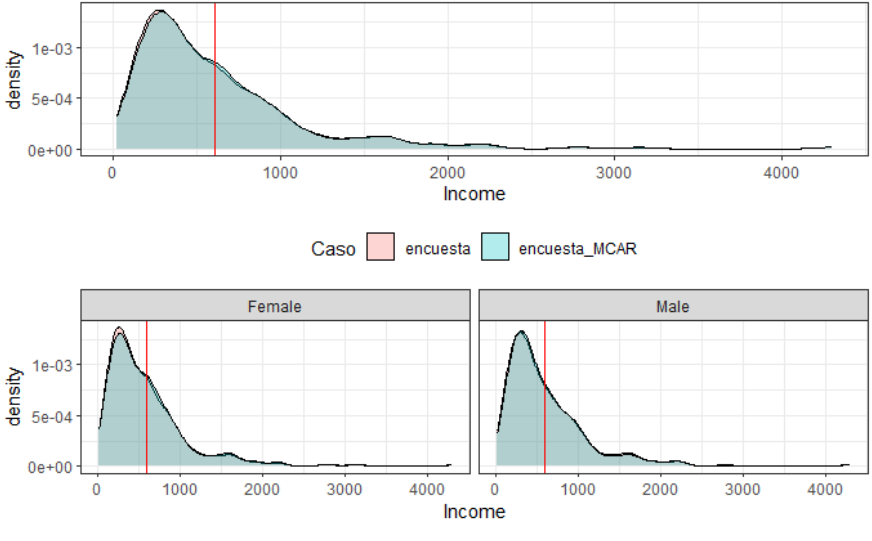


Figura 2: Densidades del ingreso con Valores perdidos por el esquema MCAR

Creando valores perdidos

```
p1 <- ggplot(dat_plot, aes(x = Expenditure, fill = Caso)) +
  geom_density(alpha = 0.3) + theme_bw() +
  theme(legend.position = "bottom") +
  geom_vline(xintercept = mean(encuesta$Expenditure),
             col = "red")

p2 <- ggplot(dat_plot, aes(x = Expenditure, fill = Caso)) +
  geom_density(alpha = 0.3) + facet_grid(.~Sex) +
  theme_bw()+
  geom_vline(xintercept = mean(encuesta$Expenditure),
             col = "red") +
  theme(legend.position = "none")
(p1/p2)
```

Creando valores perdidos

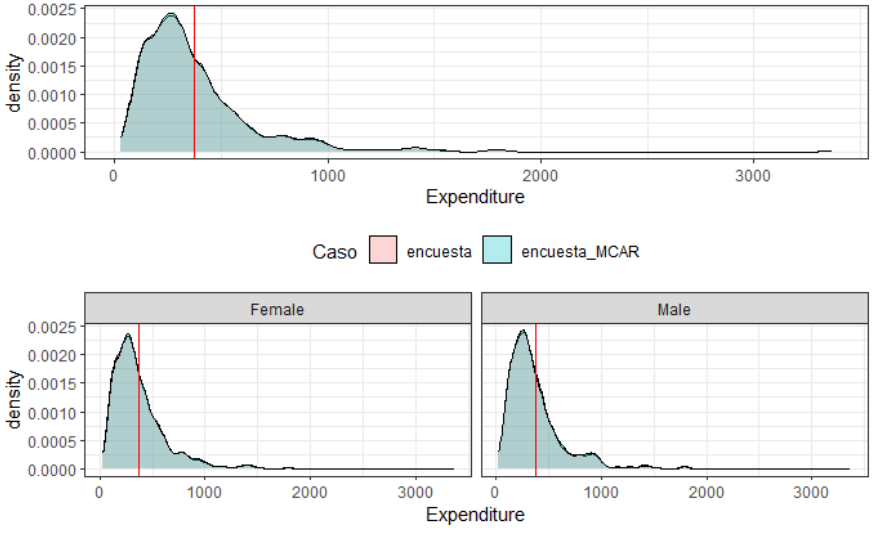


Figura 3: Densidades del gasto con valores perdidos por el esquema MCAR

Creando valores perdidos

simulemos ahora una pérdida de información MAR como sigue:

```
library(TeachingSampling)
set.seed(1234)
temp_estrato <- paste0(encuesta$Zone, encuesta$Sex)
table(temp_estrato)
```

RuralFemale	RuralMale	UrbanFemale	UrbanMale
481	428	531	439

```
sel <- S.STSI(S = temp_estrato,
              Nh = c(469,411,510,390),
              nh = c(20, 380, 20,280))
encuesta_MAR <- encuesta[-sel,]
dat_plot2 <- bind_rows(
  list(encuesta_MAR = encuesta_MAR,
       encuesta = encuesta), .id = "Caso"  )
```

Creando valores perdidos

```
p1 <- ggplot(dat_plot2, aes(x= Caso, y = Expenditure)) +
   geom_hline(yintercept = mean(encuesta$Expenditure),
              col = "red") +
  geom_boxplot() +
  facet_grid(Zone~Sex) + theme_bw()

p1
```

Creando valores perdidos

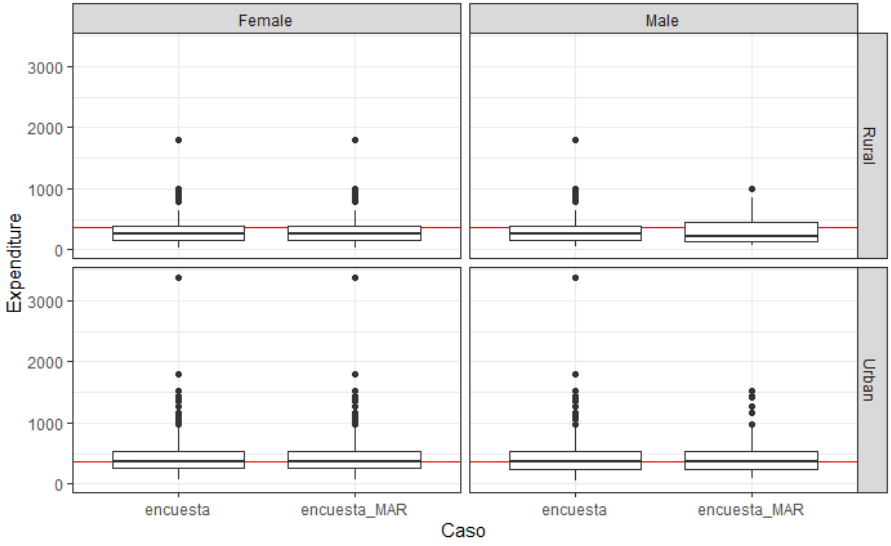


Figura 4: Valores perdidos con el esquema MCAR

Creando valores perdidos

```
p1 <- ggplot(dat_plot2, aes(x = Income, fill = Caso)) +
 geom_density(alpha = 0.3) + theme_bw() +
  theme(legend.position = "bottom") +
  geom_vline(xintercept = mean(encuesta$Income),
             col = "red")

p2 <- ggplot(dat_plot2, aes(x = Income, fill = Caso)) +
  facet_grid(.~Sex) +
  geom_density(alpha = 0.3) + theme_bw() +
  theme(legend.position = "none") +
  geom_vline(xintercept = mean(encuesta$Income),
             col = "red")
p1/p2
```

Creando valores perdidos

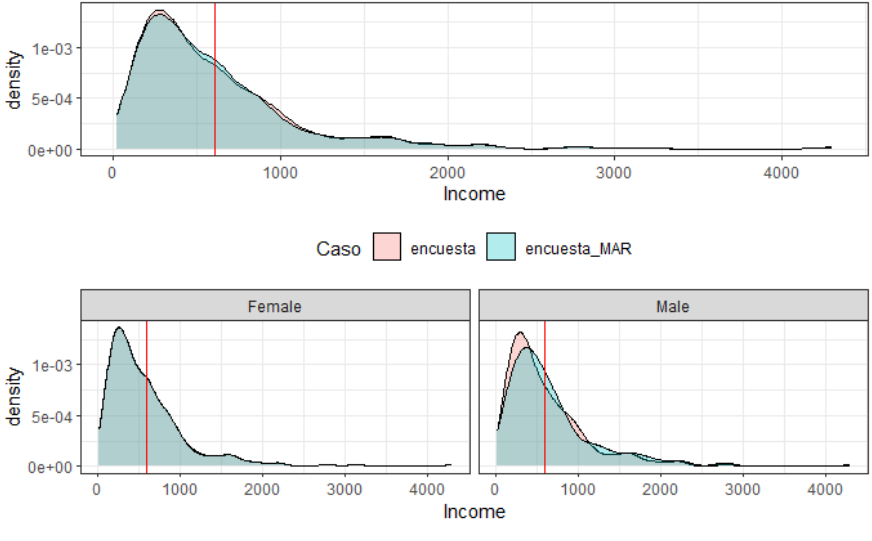


Figura 5: Densidades del ingreso con valores perdidos por el esquema MCAR

Creando valores perdidos

```
p1 <- ggplot(dat_plot2,
             aes(x = Expenditure, fill = Caso)) +
 geom_density(alpha = 0.3) + theme_bw() +
  theme(legend.position = "bottom") +
  geom_vline(
    xintercept = mean(encuesta$Expenditure),
             col = "red")

p2 <- ggplot(dat_plot2,
             aes(x = Expenditure, fill = Caso)) +
  facet_grid(.~Sex) +
  geom_density(alpha = 0.3) + theme_bw() +
  theme(legend.position = "none") +
  geom_vline(
    xintercept = mean(encuesta$Expenditure),
             col = "red")
p1/p2
```

Creando valores perdidos

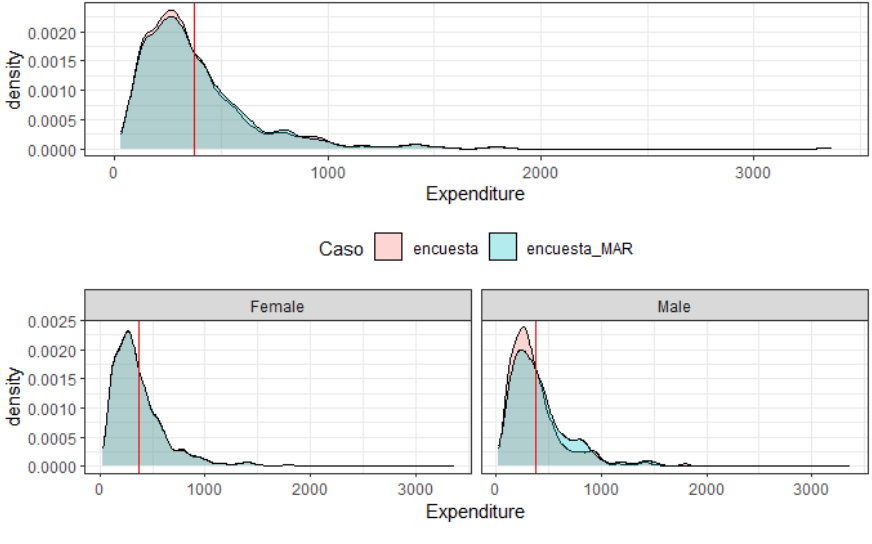


Figura 6: Densidades del gastos con valores perdidos por el esquema MCAR

Creando valores perdidos

Generemos ahora un esquema de pérdida de información en una encuesta NMAR (siglas en inglés de “Not Missing at Random”).

```
encuesta_MNAR <- encuesta %>%
  arrange((Income)) %>%
  slice(1:1300L)

dat_plot3 <- bind_rows(
  list(encuesta_MNAR = encuesta_MNAR,
       encuesta = encuesta), .id = "Caso"  )
```

Creando valores perdidos

```
p1 <- ggplot(dat_plot3, aes(x = Income, fill = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
   theme(legend.position = "bottom") +
  geom_vline(
    xintercept = mean(encuesta$Income),
               col = "red") +
  geom_vline(
    xintercept = mean(encuesta_MNAR$Income),
               col = "blue")
p1
```

Creando valores perdidos

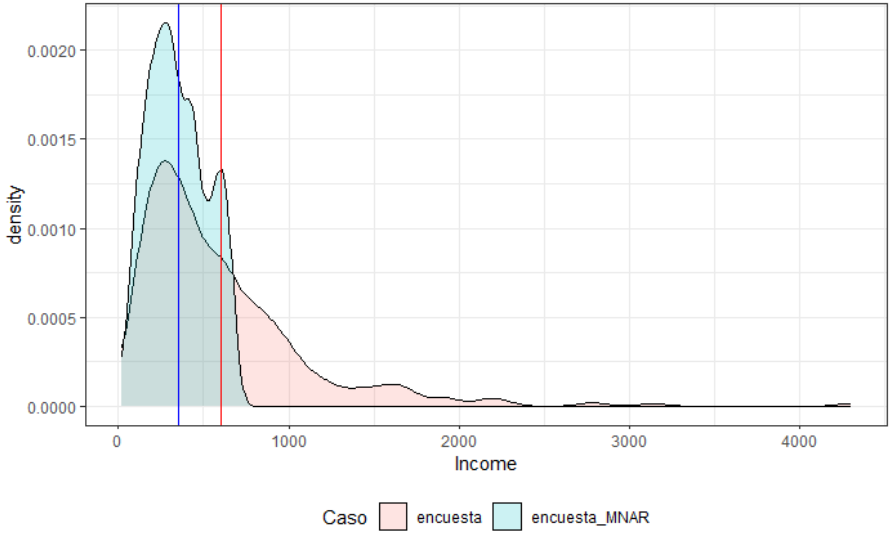


Figura 7: Densidades del ingreso con valores perdidos por el esquema MNAR

Creando valores perdidos

```
p1 <- ggplot(dat_plot3,
             aes(x = Expenditure, fill = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
   theme(legend.position = "bottom") +
  geom_vline(
    xintercept = mean(encuesta$Expenditure),
             col = "red") +
  geom_vline(
    xintercept = mean(encuesta_MNAR$Expenditure),
             col = "blue")
p1
```

Creando valores perdidos

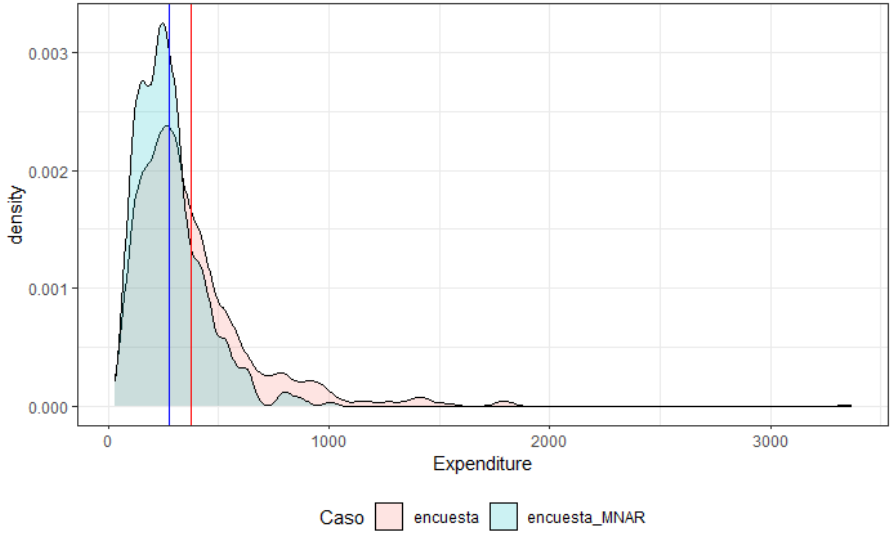


Figura 8: Densidades del gasto con valores perdidos por el esquema MNAR

Creando valores perdidos

```
p1 <- ggplot(dat_plot3, aes(x= Caso, y = Income)) +
   geom_hline(yintercept = mean(encuesta$Income),
              col = "red") +  geom_boxplot() +
  facet_grid(Zone~Sex) + theme_bw()

p1
```

Creando valores perdidos

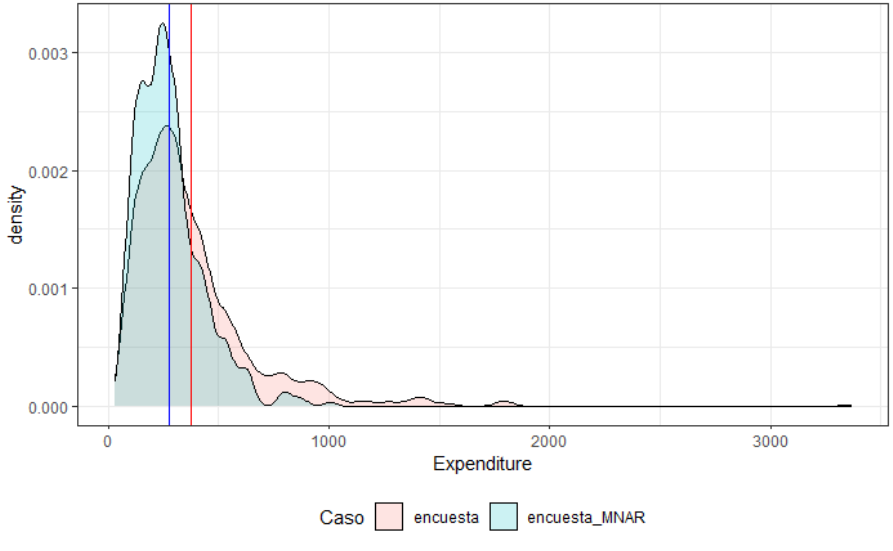


Figura 9: Impacto de la no respuesta con un esquema NMAR

Creando valores perdidos

Para efectos de ejemplificar la solución del problema a los datos faltantes en una encuesta de hogares, generemos la siguiente base de datos:

```
encuesta <- full_join(
  encuesta,
  encuesta_MCAR %>%
    select(HHID, PersonID, Income, Employment) %>%
    mutate(
      Income_missin = Income,
      Employment_missin = Employment,
      Employment = NULL,
      Income = NULL
    )
)
```

Imputación de valores perdidos.

Imputación de valores perdidos.

Para tener como referencia el porcentaje de datos faltantes, se ejecuta el siguiente comando:

```
encuesta %>% group_by(Zone) %>%
  summarise(Income = sum(is.na(Income_missin) / n()))
```

Zone	Income
Rural	0.2079
Urban	0.1928

```
encuesta %>% group_by(Sex) %>%
  summarise(Income = sum(is.na(Income_missin) / n()))
```

Sex	Income
Female	0.1838
Male	0.2191

Imputación por la media no condicional.

- Consiste en asignar el promedio de la totalidad de los datos a los valores faltantes, este método no afecta el promedio, pero si afecta la variabilidad, el sesgo y los percentiles.
- Este método es bastante simple y rápido, y puede ser útil en ciertas situaciones, especialmente cuando la variable en cuestión no tiene una distribución muy sesgada o cuando los valores faltantes son relativamente pocos en comparación con el tamaño de la muestra.

Imputación por la media no condicional.

La imputación se realiza utilizando la media aritmética de los valores no faltantes en Income_missin y se almacena en una nueva variable llamada Income_imp

```
promedio <- mean(encuesta$Income_missin, na.rm = TRUE)
encuesta %<>%
  mutate(
    Income_imp = ifelse(is.na(Income_missin),
                        promedio, Income_missin))
sum(is.na(encuesta$Income_imp))
```

```
[1] 0
```

Imputación por la media no condicional.

```
## Ordenando la base para gráfica
dat_plot4 <- tidyr::gather(
  encuesta %>% select(Zone,Sex,Income, Income_imp),
  key = "Caso", value = "Income2", -Zone,-Sex)

p1 <- ggplot(dat_plot4, aes(x = Income2, fill = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
   theme(legend.position = "bottom") +
  geom_vline(
    xintercept = mean(encuesta$Income),
               col = "red") +
  geom_vline(
    xintercept = mean(encuesta$Income_imp),
               col = "blue")
p1
```

Imputación por la media no condicional.

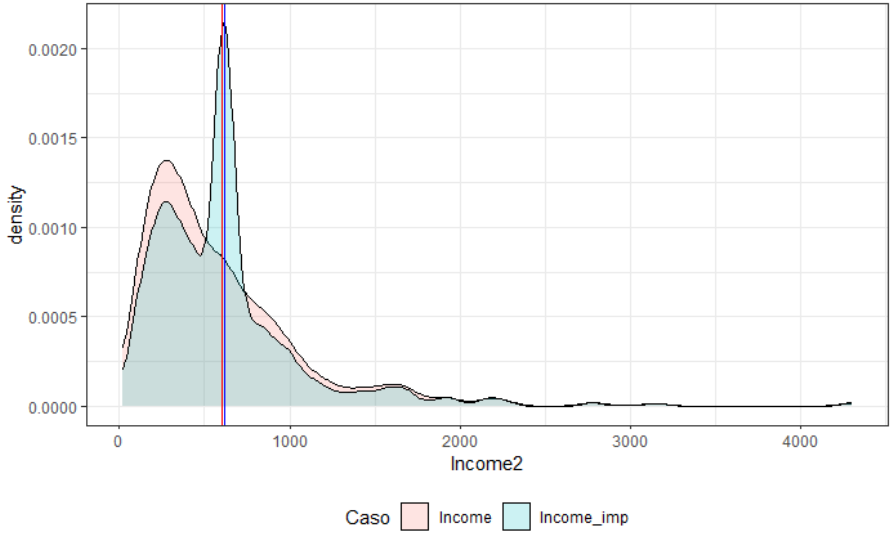


Figura 10: Imputación por la media no condicional

Imputación por la media condicional.

- A diferencia de la imputación por la media no condicional, considera otras variables al calcular la media, reconociendo que esta puede variar según los valores de otras variables.
- Se basa en la idea de que la media de una variable puede variar según los valores de otras variables en el conjunto de datos.
- Proporciona imputaciones más precisas al tener en cuenta las relaciones entre diferentes variables en el conjunto de datos.
- Es especialmente útil cuando hay correlaciones entre variables o patrones de valores faltantes en los datos.
- Aunque puede mejorar la precisión, puede ser más complejo y requerir más recursos computacionales que la imputación no condicional.

Imputación por la media condicional.

```
encuesta %<>% group_by(Stratum) %>%
  mutate(
    Income_imp = ifelse(is.na(Income_missin),
     mean(Income_missin, na.rm = TRUE),
     Income_missin)) %>% data.frame()
sum(is.na(encuesta$Income_imp))
```

```
[1] 0
```

```
encuesta %<>%
  mutate(
    Income_imp = ifelse(is.na(Income_imp),
                        promedio, Income_imp))
sum(is.na(encuesta$Income_imp))
```

```
[1] 0
```

Imputación por la media condicional.

Se calculan las medias y desviaciones estándar tanto para los datos imputados como los originales y así poder comparar le efecto de la imputación realizada:

```
encuesta %>% summarise(
  Income_ = mean(Income),
  Income_sd = sd(Income),
  Income_imp_ = mean(Income_imp),
  Income_imp_sd = sd(Income_imp))
```

Income_	Income_sd	Income_imp_	Income_imp_sd
604.2	513.1	611.5	488.7

Para poder comparar los resultados, calculemos el sesgo relativo de la imputación el cual se calcula como sigue:

$$BR = \frac{Income - Income_{imp}}{Income} \times 100\%$$

```
100*(604.2- 611.5   )/604.2
```

```
[1] -1.208
```

Imputación por la media condicional.

```
encuesta %>%group_by(Zone) %>%  summarise(
  Income_ = mean(Income),
  Income_sd = sd(Income),
  Income_imp_ = mean(Income_imp),
  Income_imp_sd = sd(Income_imp)) %>%
  mutate(BR = 100*(Income_ - Income_imp_)/Income_ )
```

Zone	Income_	Income_sd	Income_imp_	Income_imp_sd	BR
Rural	469.1	336.6	477.9	305.5	-1.8721
Urban	730.9	609.0	736.8	585.7	-0.8075

Imputación por la media condicional.

```
encuesta %>%group_by(Sex) %>%  summarise(
  Income_ = mean(Income),
  Income_sd = sd(Income),
  Income_imp_ = mean(Income_imp),
  Income_imp_sd = sd(Income_imp)) %>%
  mutate(BR = 100*(Income_ - Income_imp_)/Income_ )
```

Sex	Income_	Income_sd	Income_imp_	Income_imp_sd	BR
Female	589.2	504.3	600.3	486.6	-1.8822
Male	621.8	522.9	624.6	491.2	-0.4609

Imputación por la media condicional.

```
## Ordenando la base para gráfica
dat_plot5 <- tidyr::gather(
  encuesta %>% select(Zone,Sex,Income, Income_imp),
  key = "Caso", value = "Income2", -Zone,-Sex)

p1 <- ggplot(dat_plot5, aes(x = Income2, fill = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
   theme(legend.position = "bottom") +
  geom_vline(
    xintercept = mean(encuesta$Income),
               col = "red") +
  geom_vline(
    xintercept = mean(encuesta$Income_imp),
               col = "blue")
p1
```

Imputación por la media condicional.

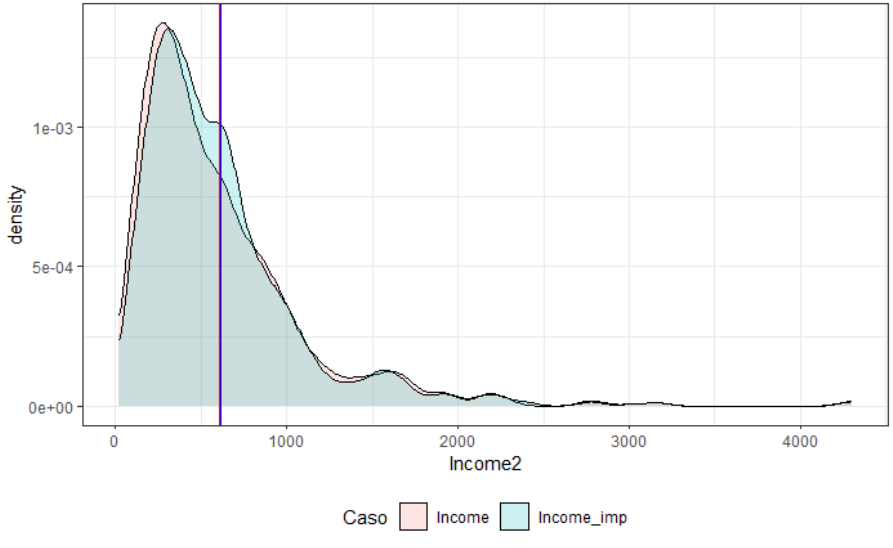


Figura 11: Imputación por la media no condicional sobre el ingreso

Imputación por la media condicional.

Si se observa ahora la distribunción de los datos por zona y sexo, se puede observar también una buena imputación de las observaciones.

```
p1 <- ggplot(dat_plot5, aes(x= Caso, y = Income2)) +
   geom_hline(yintercept = mean(encuesta$Income),
              col = "red") +  geom_boxplot() +
  facet_grid(Zone~Sex) + theme_bw()
p1
```

Imputación por la media condicional.

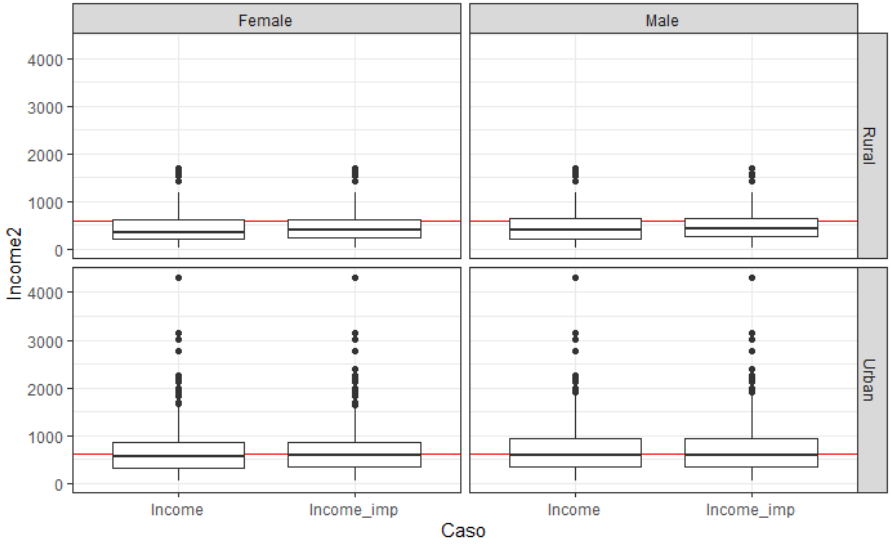


Figura 12: Imputación por la media no condicional Sexo y Zona

Imputación por Hot-deck y Cold-deck

Hot-deck La imputación *hot deck* consiste en reemplazar los valores faltantes de una o más variables para un no encuestado (llamado receptor) con valores observados de un encuestado (el donante) que es similar al no encuestado con respecto a las características observadas en ambos casos.

Cold-deck A este método lo llamamos *Cold-deck* por analogía con *Hot-deck*. El método consiste en reemplazar el valor faltante por valores de una fuente no relacionada con el conjunto de datos en consideración. Por ejemplo, se pide a un grupo de personas diligenciar una cuestionario sobre hábitos de lectura y que cinco personas no respondieron a un ítem. Entonces, la imputación de la respuesta por *Cold-deck* es sustituir las respuestas con información de un donante similar en una encuesta realizada anteriormente.

Imputación por hot-deck

```
donante <- which(!is.na(encuesta$Income_missin))
receptor <- which(is.na(encuesta$Income_missin))
encuesta$Income_imp <- encuesta$Income_missin
set.seed(1234)
for(ii in receptor){
don_ii <- sample(x = donante, size = 1)
encuesta$Income_imp[ii] <-
  encuesta$Income_missin[don_ii]
}
sum(is.na(encuesta$Income_imp))
```

```
[1] 0
```

Imputación por hot-deck

Una vez realizada la imputación, se calcula la media y la desviación de los datos completos e imputados:

```
encuesta %>% summarise(
  Income_ = mean(Income),
  Income_sd = sd(Income),
  Income_imp_ = mean(Income_imp),
  Income_imp_sd = sd(Income_imp)) %>%
  mutate(BR = 100*(Income_ - Income_imp_)/Income_ )
```

Income_	Income_sd	Income_imp_	Income_imp_sd	BR
604.2	513.1	618.3	528.2	-2.324

Imputación por hot-deck

Una vez realizada la imputación, se calcula la media y la desviación de los datos completos e imputados:

```
encuesta %>%group_by(Zone) %>%  summarise(
  Income_ = mean(Income),
  Income_sd = sd(Income),
  Income_imp_ = mean(Income_imp),
  Income_imp_sd = sd(Income_imp))%>%
  mutate(BR = 100*(Income_ - Income_imp_)/Income_ )
```

Zone	Income_	Income_sd	Income_imp_	Income_imp_sd	BR
Rural	469.1	336.6	503.7	368.9	-7.3736
Urban	730.9	609.0	725.7	624.0	0.7129

Imputación por hot-deck

```
encuesta %>%group_by(Sex) %>%  summarise(
  Income_ = mean(Income),
  Income_sd = sd(Income),
  Income_imp_ = mean(Income_imp),
  Income_imp_sd = sd(Income_imp)) %>%
  mutate(BR = 100*(Income_ - Income_imp_)/Income_ )
```

Sex	Income_	Income_sd	Income_imp_	Income_imp_sd	BR
Female	589.2	504.3	602.8	503.1	-2.304
Male	621.8	522.9	636.4	555.9	-2.347

Imputación por hot-deck

```
## Ordenando la base para gráfica
dat_plot6 <- tidyr::gather(
  encuesta %>% select(Zone,Sex,Income, Income_imp),
  key = "Caso", value = "Income2", -Zone,-Sex)

p1 <- ggplot(dat_plot6, aes(x = Income2, fill = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
   theme(legend.position = "bottom") +
  geom_vline(
    xintercept = mean(encuesta$Income),
               col = "red") +
  geom_vline(
    xintercept = mean(encuesta$Income_imp),
               col = "blue")
p1
```

Imputación por hot-deck

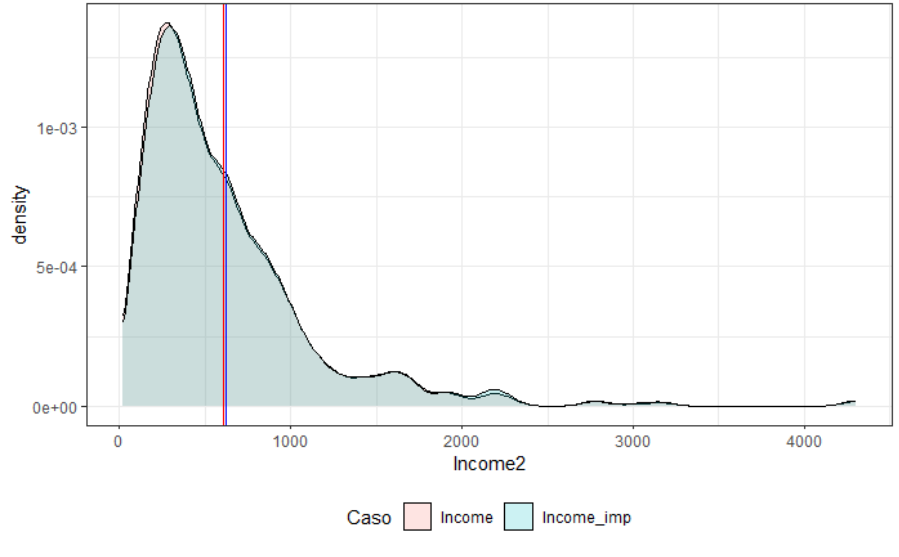


Figura 13: Imputación por hot-deck

Imputación por hot-deck

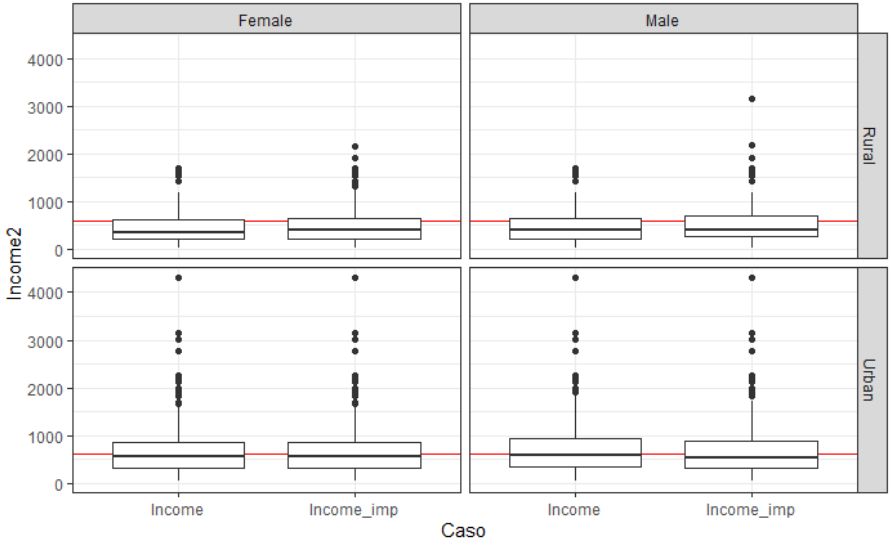


Figura 14: Imputación por hot-deck por sexo y zona

Imputación por hot-deck

Se implementa el método de imputación pero para la variable empleado

```
donante <- which(!is.na(encuesta$Income_missin))
receptor <- which(is.na(encuesta$Income_missin))
encuesta$Employment_imp <- encuesta$Employment_missin
(prop <- prop.table(
  table(na.omit(encuesta$Employment_missin))))
```

Unemployed	Inactive	Employed
0.0426	0.3739	0.5835

Imputación por hot-deck estado de ocupación

```
set.seed(1234)
imp <- sample(size = length(receptor),
  c("Unemployed", "Inactive","Employed"),
       prob = prop, replace = TRUE     )
encuesta$Employment_imp[receptor] <- imp
sum(is.na(encuesta$Employment_imp))
```

```
[1] 0
```

Imputación por hot-deck

Resultados antes de la imputación

```
prop.table(
  table(encuesta$Employment_missin, useNA = "a"))
```

Unemployed	Inactive	Employed	NA
0.0341	0.2991	0.4667	0.2001

Resultados después de la imputación

```
prop.table(
  table(encuesta$Employment_imp, useNA = "a"))
```

Unemployed	Inactive	Employed	NA
0.0442	0.3672	0.5886	0

Imputación por hot-deck

Resultados antes de la imputación

Zone	Unemployed	Inactive	Employed	Sum	<NA>
Rural	0.0117	0.1506	0.2209	0.4838	0.1006
Urban	0.0224	0.1485	0.2459	0.5162	0.0995
Sum	0.0341	0.2991	0.4667	1.0000	0.2001
NA	0.0000	0.0000	0.0000	0.0000	0.0000

Resultados después de la imputación

Zone	Unemployed	Inactive	Employed	Sum	<NA>
Rural	0.0160	0.1847	0.2831	0.4838	0
Urban	0.0282	0.1825	0.3055	0.5162	0
Sum	0.0442	0.3672	0.5886	1.0000	0
NA	0.0000	0.0000	0.0000	0.0000	0

Imputación por hot-deck

Resultados antes de la imputación

Sex	Unemployed	Inactive	Employed	Sum	<NA>
Female	0.0106	0.2278	0.2012	0.5386	0.0990
Male	0.0234	0.0713	0.2656	0.4614	0.1011
Sum	0.0341	0.2991	0.4667	1.0000	0.2001
NA	0.0000	0.0000	0.0000	0.0000	0.0000

Resultados después de la imputación

Sex	Unemployed	Inactive	Employed	Sum	<NA>
Female	0.0149	0.2650	0.2586	0.5386	0
Male	0.0293	0.1022	0.3300	0.4614	0
Sum	0.0442	0.3672	0.5886	1.0000	0
NA	0.0000	0.0000	0.0000	0.0000	0

Imputación por regresión

- La imputación por regresión es una técnica que estima y asigna valores a datos faltantes basándose en un modelo de regresión construido a partir de variables disponibles en el conjunto de datos.
- Se selecciona una variable objetivo con valores faltantes y se identifican variables predictoras con correlación significativa. Se ajusta un modelo de regresión utilizando estas variables para predecir los valores faltantes.
- Requiere conocimientos sólidos de análisis de datos y modelado estadístico. Su aplicación puede depender de la calidad y cantidad de datos disponibles y la distribución de los valores faltantes.
- **Limitaciones:** Debe utilizarse con precaución y considerando sus limitaciones, ya que su eficacia depende de la validez del modelo y la disponibilidad de datos.

Imputación por regresión

- Se ejemplifica imputando las variables de ingreso y empleados utilizando modelos de regresión lineal múltiple y multinomial, respectivamente, con covariables como zona, sexo y empleo.

```
require(nnet)
encuesta$Income_imp <- encuesta$Income_missin
encuesta$Employment_imp <- encuesta$Employment_missin
encuesta_obs <- filter(encuesta,
                       !is.na(Income_missin))
encuesta_no_obs <- filter(encuesta,
                          is.na(Income_missin))
```

Imputación por regresión

Modelo para imputación del ingreso

```
mod <- lm(Income~Zone + Sex +Expenditure,
          data = encuesta_obs)
```

Modelo para imputación del estado de ocupación

```
mod.mult <- multinom(
             Employment~Zone + Sex +Expenditure,
             data = encuesta_obs)
```

```
# weights:  15 (8 variable)
initial  value 1651.214270
iter  10 value 1182.110113
final  value 1132.682019
converged
```

Imputación por regresión

Una vez ajustado los modelos tanto para las variable ingreso como para empleados, se realiza el proceso de predicción como se muestra a continuación:

```
imp <- predict(mod, encuesta_no_obs)
imp.mult <- predict(mod.mult, encuesta_no_obs,
                    type =  "class")
encuesta_no_obs$Income_imp <- imp
encuesta_no_obs$Employment_imp <- imp.mult
encuesta <- bind_rows(encuesta_obs,encuesta_no_obs)
```

Imputación por regresión

Resultados antes de la imputación

```
prop.table(
  table(encuesta$Employment_missin, useNA = "a"))
```

Unemployed	Inactive	Employed	NA
0.0341	0.2991	0.4667	0.2001

Resultados después de la imputación

```
prop.table(
  table(encuesta$Employment_imp, useNA = "a"))
```

Unemployed	Inactive	Employed	NA
0.0341	0.3858	0.5801	0

Imputación por regresión

Resultados antes de la imputación

Zone	Unemployed	Inactive	Employed	Sum	<NA>
Rural	0.0117	0.1506	0.2209	0.4838	0.1006
Urban	0.0224	0.1485	0.2459	0.5162	0.0995
Sum	0.0341	0.2991	0.4667	1.0000	0.2001
NA	0.0000	0.0000	0.0000	0.0000	0.0000

Resultados después de la imputación

Zone	Unemployed	Inactive	Employed	Sum	<NA>
Rural	0.0117	0.2006	0.2714	0.4838	0
Urban	0.0224	0.1852	0.3087	0.5162	0
Sum	0.0341	0.3858	0.5801	1.0000	0
NA	0.0000	0.0000	0.0000	0.0000	0

Imputación por regresión

Resultados antes de la imputación

Sex	Unemployed	Inactive	Employed	Sum	<NA>
Female	0.0106	0.2278	0.2012	0.5386	0.0990
Male	0.0234	0.0713	0.2656	0.4614	0.1011
Sum	0.0341	0.2991	0.4667	1.0000	0.2001
NA	0.0000	0.0000	0.0000	0.0000	0.0000

Resultados después de la imputación

Sex	Unemployed	Inactive	Employed	Sum	<NA>
Female	0.0106	0.3145	0.2134	0.5386	0
Male	0.0234	0.0713	0.3667	0.4614	0
Sum	0.0341	0.3858	0.5801	1.0000	0
NA	0.0000	0.0000	0.0000	0.0000	0

Imputación por regresión

```
encuesta %>% summarise(
  Income_ = mean(Income),
  Income_sd = sd(Income),
  Income_imp_ = mean(Income_imp),
  Income_imp_sd = sd(Income_imp))  %>%
  mutate(BR = 100*(Income_ - Income_imp_)/Income_ )
```

Income_	Income_sd	Income_imp_	Income_imp_sd	BR
604.2	513.1	611.7	498.3	-1.241

Imputación por regresión

```
encuesta %>%group_by(Zone) %>%  summarise(
  Income_ = mean(Income),
  Income_sd = sd(Income),
  Income_imp_ = mean(Income_imp),
  Income_imp_sd = sd(Income_imp))  %>%
  mutate(BR = 100*(Income_ - Income_imp_)/Income_ )
```

Zone	Income_	Income_sd	Income_imp_	Income_imp_sd	BR
Rural	469.1	336.6	476.1	317.4	-1.495
Urban	730.9	609.0	738.8	594.5	-1.088

Imputación por regresión

```
encuesta %>%group_by(Sex) %>%  summarise(
  Income_ = mean(Income),
  Income_sd = sd(Income),
  Income_imp_ = mean(Income_imp),
  Income_imp_sd = sd(Income_imp))  %>%
  mutate(BR = 100*(Income_ - Income_imp_)/Income_ )
```

Sex	Income_	Income_sd	Income_imp_	Income_imp_sd	BR
Female	589.2	504.3	598.7	488.4	-1.5984
Male	621.8	522.9	627.0	509.5	-0.8455

Imputación por regresión

```
## Ordenando la base para gráfica
dat_plot7 <- tidyr::gather(
  encuesta %>% select(Zone,Sex,Income, Income_imp),
  key = "Caso", value = "Income2", -Zone,-Sex)

p1 <- ggplot(dat_plot7, aes(x = Income2, fill = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
   theme(legend.position = "bottom") +
  geom_vline(
    xintercept = mean(encuesta$Income),
               col = "red") +
  geom_vline(
    xintercept = mean(encuesta$Income_imp),
               col = "blue")
p1
```

Imputación por regresión

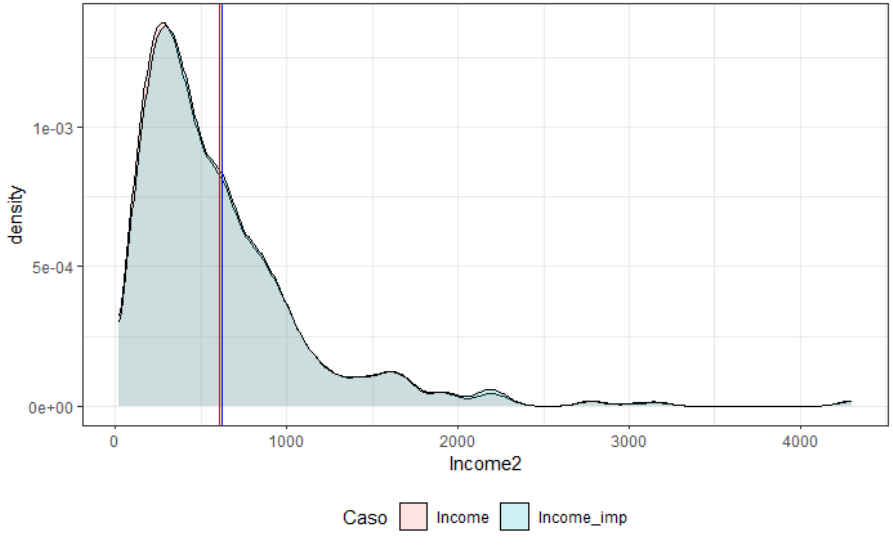


Figura 15: Imputación por regresión

Imputación por regresión

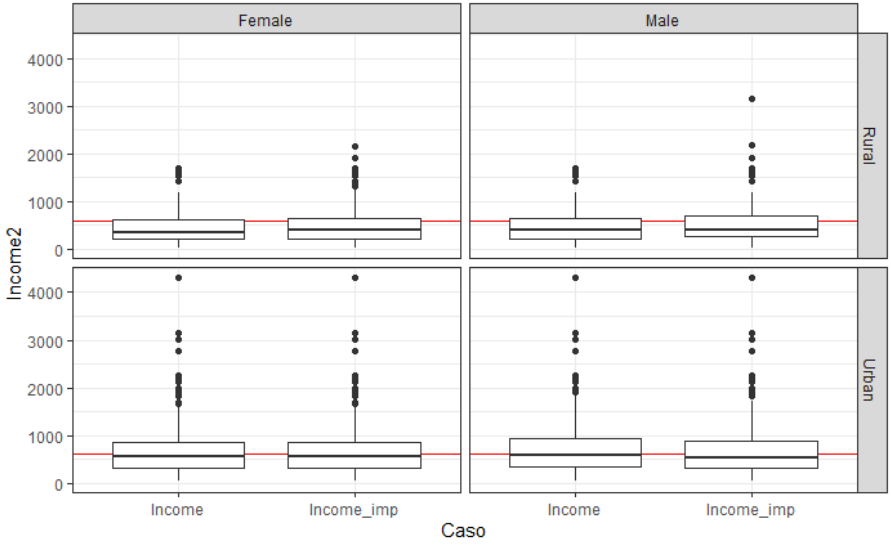


Figura 16: Imputación por regresión por sexo y zona

Imputación por el vecino más cercano

- La imputación por el vecino más cercano es una técnica que reemplaza valores faltantes en un conjunto de datos utilizando valores de observaciones similares en función de ciertas variables.
- Se basa en la premisa de que registros similares tienden a tener valores similares para una variable específica.
- **Proceso:**
  1. **Definición de Magnitud de Distancia:** Se elige una medida de distancia, como la euclidiana o la de Manhattan.
  2. **Identificación del Donante:** Para cada elemento con valor faltante, se identifica el registro donante más cercano en función de la distancia definida.
  3. **Imputación:** Se sustituye el valor faltante con la información del donante identificado.

Imputación por el vecino más cercano

- **Valor de k:** Representa el número de vecinos más cercanos utilizados en la estimación.
- **Medida de Distancia:** La elección de la métrica de distancia afecta los resultados.
- Es una técnica simple y fácil de implementar, pero su eficacia depende de la cantidad y calidad de los datos y de la elección adecuada de parámetros.
- Antes de utilizar la técnica, es crucial evaluar la calidad de los datos y los resultados obtenidos para garantizar la validez de la imputación.

Imputación por el vecino más cercano

- Crear nuevas columnas para imputar valores faltantes

```
encuesta$Income_imp <- encuesta$Income_missin
encuesta$Employment_imp <- encuesta$Employment_missin
```

- Filtrar observaciones con valores faltantes en la variable 'Income_missin'

```
encuesta_obs <- filter(encuesta,
                       !is.na(Income_missin))
```

- Filtrar observaciones sin valores en la variable 'Income_missin' (valores faltantes)

```
encuesta_no_obs <- filter(encuesta,
                          is.na(Income_missin))
```

Imputación por el vecino más cercano

Iterar sobre cada fila de la encuesta_no_obs

```
for(ii in 1:nrow(encuesta_no_obs)){
  # Obtener el valor de Expenditure en la fila actual
  Expen_ii <- encuesta_no_obs$Expenditure[[ii]]

  # Encontrar el índice del valor más cercano en encuesta_obs
  don_ii <- which.min(abs(Expen_ii - encuesta_obs$Expenditure))

  # Asignar el valor de Income_missin correspondiente al índice encontrado
  encuesta_no_obs$Income_imp[[ii]] <- encuesta_obs$Income_missin[[don_ii]]

  # Asignar el valor de Employment_missin correspondiente al índice encontrado
  encuesta_no_obs$Employment_imp[[ii]] <-
    encuesta_obs$Employment_missin[[don_ii]]
}

# Combinar encuesta_obs y encuesta_no_obs en un solo dataframe
encuesta <- bind_rows(encuesta_obs,encuesta_no_obs)
```

Imputación por el vecino más cercano

Resultados antes de la imputación

```
prop.table(
  table(encuesta$Employment_missin, useNA = "a"))
```

Unemployed	Inactive	Employed	NA
0.0341	0.2991	0.4667	0.2001

Resultados después de la imputación

```
prop.table(
  table(encuesta$Employment_imp, useNA = "a"))
```

Unemployed	Inactive	Employed	NA
0.0436	0.3651	0.5913	0

Imputación por el vecino más cercano

Resultados antes de la imputación

Zone	Unemployed	Inactive	Employed	Sum	<NA>
Rural	0.0117	0.1506	0.2209	0.4838	0.1006
Urban	0.0224	0.1485	0.2459	0.5162	0.0995
Sum	0.0341	0.2991	0.4667	1.0000	0.2001
NA	0.0000	0.0000	0.0000	0.0000	0.0000

Resultados después de la imputación

Zone	Unemployed	Inactive	Employed	Sum	<NA>
Rural	0.0128	0.1873	0.2837	0.4838	0
Urban	0.0309	0.1778	0.3076	0.5162	0
Sum	0.0436	0.3651	0.5913	1.0000	0
NA	0.0000	0.0000	0.0000	0.0000	0

Imputación por el vecino más cercano

Resultados antes de la imputación

Sex	Unemployed	Inactive	Employed	Sum	<NA>
Female	0.0106	0.2278	0.2012	0.5386	0.0990
Male	0.0234	0.0713	0.2656	0.4614	0.1011
Sum	0.0341	0.2991	0.4667	1.0000	0.2001
NA	0.0000	0.0000	0.0000	0.0000	0.0000

Resultados después de la imputación

Sex	Unemployed	Inactive	Employed	Sum	<NA>
Female	0.0160	0.2528	0.2698	0.5386	0
Male	0.0277	0.1123	0.3214	0.4614	0
Sum	0.0436	0.3651	0.5913	1.0000	0
NA	0.0000	0.0000	0.0000	0.0000	0

Imputación por el vecino más cercano

```
encuesta %>% summarise(
  Income_ = mean(Income),
  Income_sd = sd(Income),
  Income_imp_ = mean(Income_imp),
  Income_imp_sd = sd(Income_imp))%>%
  mutate(BR = 100*(Income_ - Income_imp_)/Income_ )
```

Income_	Income_sd	Income_imp_	Income_imp_sd	BR
604.2	513.1	610.5	513.7	-1.035

Imputación por el vecino más cercano

```
encuesta %>%group_by(Zone) %>%  summarise(
  Income_ = mean(Income),
  Income_sd = sd(Income),
  Income_imp_ = mean(Income_imp),
  Income_imp_sd = sd(Income_imp))%>%
  mutate(BR = 100*(Income_ - Income_imp_)/Income_ )
```

Zone	Income_	Income_sd	Income_imp_	Income_imp_sd	BR
Rural	469.1	336.6	477.9	344.1	-1.8746
Urban	730.9	609.0	734.8	607.0	-0.5304

Imputación por el vecino más cercano

```
encuesta %>%group_by(Sex) %>%  summarise(
  Income_ = mean(Income),
  Income_sd = sd(Income),
  Income_imp_ = mean(Income_imp),
  Income_imp_sd = sd(Income_imp)) %>%
  mutate(BR = 100*(Income_ - Income_imp_)/Income_ )
```

Sex	Income_	Income_sd	Income_imp_	Income_imp_sd	BR
Female	589.2	504.3	597.8	504.6	-1.4548
Male	621.8	522.9	625.3	524.0	-0.5712

Imputación por el vecino más cercano

```
## Ordenando la base para gráfica
dat_plot8 <- tidyr::gather(
  encuesta %>% select(Zone,Sex,Income, Income_imp),
  key = "Caso", value = "Income2", -Zone,-Sex)

p1 <- ggplot(dat_plot8, aes(x = Income2, fill = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
   theme(legend.position = "bottom") +
  geom_vline(
    xintercept = mean(encuesta$Income),
               col = "red") +
  geom_vline(
    xintercept = mean(encuesta$Income_imp),
               col = "blue")
p1
```

Imputación por el vecino más cercano

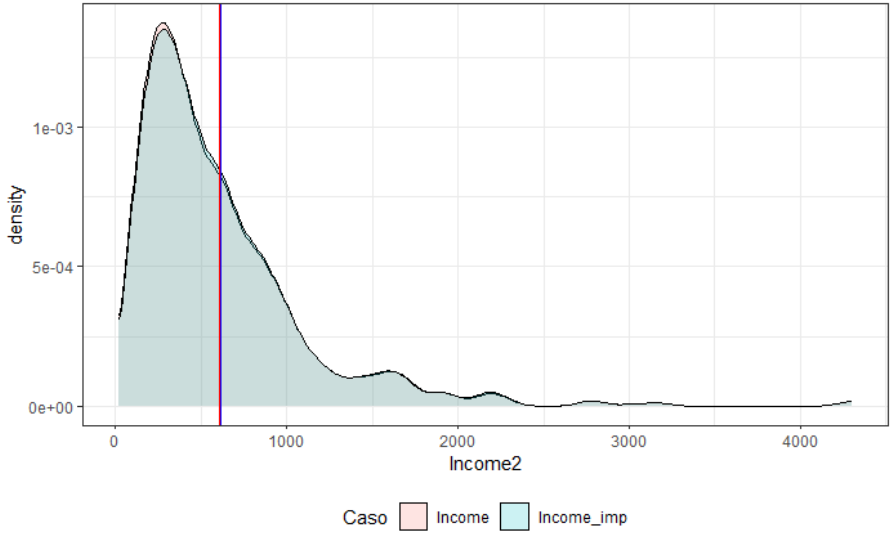


Figura 17: Imputación por el vecino más cercano

Imputación por el vecino más cercano

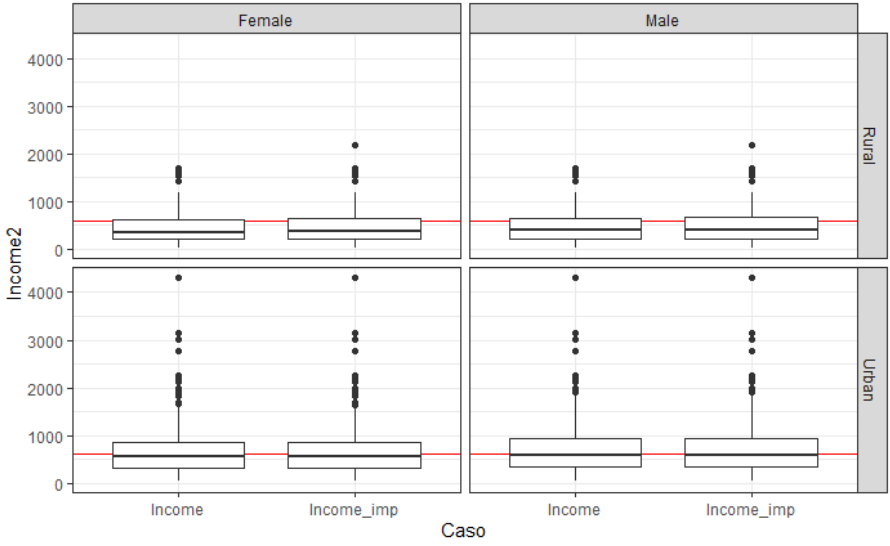


Figura 18: Imputación por el vecino más cercano por sexo y zona

Imputación por el vecino más cercano con regresión

se presentan los pasos que se deben tener en cuenta para realizar la imputación utilizando el vecino más cernado mediante una regresión:

Paso 1: Ajustar un modelo de regresión.

Paso 2: Realizar la predicción de los valores observados y no observados.

Paso 3: Comparar las predicciones obtenidas para los valores observados y no observados.

Paso 4: Para la i-ésima observación identificar el donante con la menor distancia al receptor.

Paso 5: Reemplazar el valor faltante con la información proveniente del donante.

NOTA Se toma es la información observada en el donante.

Imputación por el vecino más cercano con regresión

```r
# Imputación de valores faltantes en las columnas
# 'Income_imp' y 'Employment_imp'
encuesta$Income_imp <- encuesta$Income_missin
encuesta$Employment_imp <- encuesta$Employment_missin

# Filtrar observaciones con valores disponibles
# en la variable 'Income_missin'
encuesta_obs <- filter(encuesta,
                       !is.na(Income_missin))

# Filtrar observaciones con valores faltantes
# en la variable 'Income_missin'
encuesta_no_obs <- filter(encuesta,
                          is.na(Income_missin))

# Ajuste de un modelo de regresión lineal utilizando las observaciones con
mod <- lm(Income ~ Zone + Sex + Expenditure,
          data = encuesta_obs)
```

Imputación por el vecino más cercano con regresión

```
# Predicciones para las observaciones con 'Income_missin'
# disponibles y sin valores
pred_Obs <- predict(mod, encuesta_obs)
pred_no_Obs <- predict(mod, encuesta_no_obs)

# Imputación de valores faltantes utilizando el vecino
# más cercano en las predicciones
for (ii in 1:nrow(encuesta_no_obs)) {
  don_ii <- which.min(abs(pred_no_Obs[ii] - pred_Obs))
  encuesta_no_obs$Income_imp[[ii]] <- encuesta_obs$Income_missin[[don_ii]]
  encuesta_no_obs$Employment_imp[[ii]] <- encuesta_obs$Employment_missin[[don_ii]]
}

# Combinar las observaciones imputadas con las observaciones originales
encuesta <- bind_rows(encuesta_obs, encuesta_no_obs)
```

Imputación por el vecino más cercano con regresión

Resultados antes de la imputación

```
prop.table(
  table(encuesta$Employment_missin, useNA = "a"))
```

Unemployed	Inactive	Employed	NA
0.0341	0.2991	0.4667	0.2001

Resultados después de la imputación

```
prop.table(
  table(encuesta$Employment_imp, useNA = "a"))
```

Unemployed	Inactive	Employed	NA
0.0399	0.3736	0.5865	0

Imputación por el vecino más cercano con regresión

Resultados antes de la imputación

Zone	Unemployed	Inactive	Employed	Sum	<NA>
Rural	0.0117	0.1506	0.2209	0.4838	0.1006
Urban	0.0224	0.1485	0.2459	0.5162	0.0995
Sum	0.0341	0.2991	0.4667	1.0000	0.2001
NA	0.0000	0.0000	0.0000	0.0000	0.0000

Resultados después de la imputación

Zone	Unemployed	Inactive	Employed	Sum	<NA>
Rural	0.0138	0.1905	0.2794	0.4838	0
Urban	0.0261	0.1831	0.3071	0.5162	0
Sum	0.0399	0.3736	0.5865	1.0000	0
NA	0.0000	0.0000	0.0000	0.0000	0

Imputación por el vecino más cercano con regresión

Resultados antes de la imputación

Sex	Unemployed	Inactive	Employed	Sum	<NA>
Female	0.0106	0.2278	0.2012	0.5386	0.0990
Male	0.0234	0.0713	0.2656	0.4614	0.1011
Sum	0.0341	0.2991	0.4667	1.0000	0.2001
NA	0.0000	0.0000	0.0000	0.0000	0.0000

Resultados después de la imputación

Sex	Unemployed	Inactive	Employed	Sum	<NA>
Female	0.0122	0.2730	0.2533	0.5386	0
Male	0.0277	0.1006	0.3332	0.4614	0
Sum	0.0399	0.3736	0.5865	1.0000	0
NA	0.0000	0.0000	0.0000	0.0000	0

Imputación por el vecino más cercano con regresión

```
encuesta %>% summarise(
  Income_ = mean(Income),
  Income_sd = sd(Income),
  Income_imp_ = mean(Income_imp),
  Income_imp_sd = sd(Income_imp)) %>%
  mutate(BR = 100*(Income_ - Income_imp_)/Income_ )
```

Income_	Income_sd	Income_imp_	Income_imp_sd	BR
604.2	513.1	608.3	515.6	-0.6661

Imputación por el vecino más cercano con regresión

```
encuesta %>%group_by(Zone) %>%  summarise(
  Income_ = mean(Income),
  Income_sd = sd(Income),
  Income_imp_ = mean(Income_imp),
  Income_imp_sd = sd(Income_imp)) %>%
  mutate(BR = 100*(Income_ - Income_imp_)/Income_ )
```

Zone	Income_	Income_sd	Income_imp_	Income_imp_sd	BR
Rural	469.1	336.6	476.1	342.7	-1.4781
Urban	730.9	609.0	732.2	611.1	-0.1778

Imputación por el vecino más cercano con regresión

```
encuesta %>%group_by(Sex) %>%  summarise(
  Income_ = mean(Income),
  Income_sd = sd(Income),
  Income_imp_ = mean(Income_imp),
  Income_imp_sd = sd(Income_imp)) %>%
  mutate(BR = 100*(Income_ - Income_imp_)/Income_ )
```

Sex	Income_	Income_sd	Income_imp_	Income_imp_sd	BR
Female	589.2	504.3	593.0	508.4	-0.6380
Male	621.8	522.9	626.1	523.5	-0.6973

Imputación por el vecino más cercano con regresión

```r
## Ordenando la base para gráfica
dat_plot9 <- tidyr::gather(
  encuesta %>% select(Zone, Sex, Income, Income_imp),
  key = "Caso",
  value = "Income2",
  -Zone,
  -Sex
)

p1 <- ggplot(dat_plot9, aes(x = Income2, fill = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
  theme(legend.position = "bottom") +
  geom_vline(xintercept = mean(encuesta$Income),
             col = "red") +
  geom_vline(xintercept = mean(encuesta$Income_imp),
             col = "blue")
p1
```

Imputación por el vecino más cercano con regresión

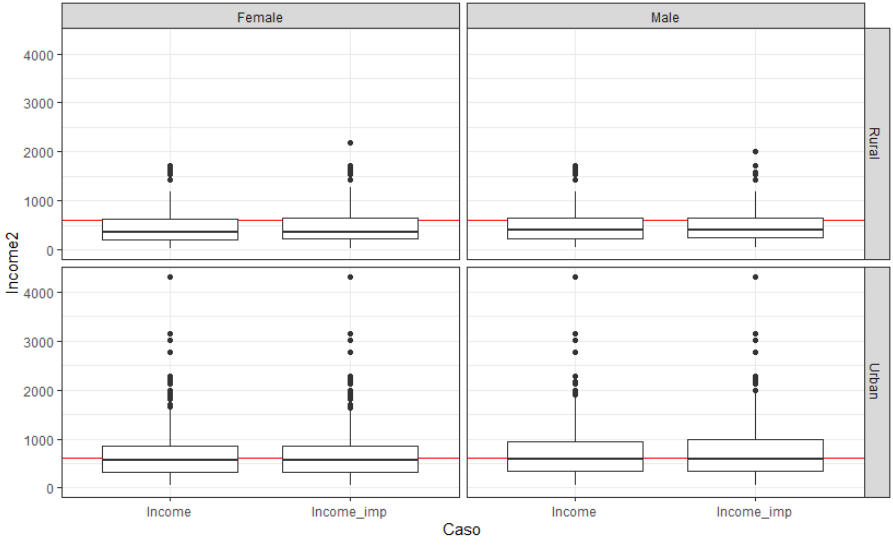


Figura 19: Imputación por el vecino más cercano con regresión

Imputación por el vecino más cercano con regresión

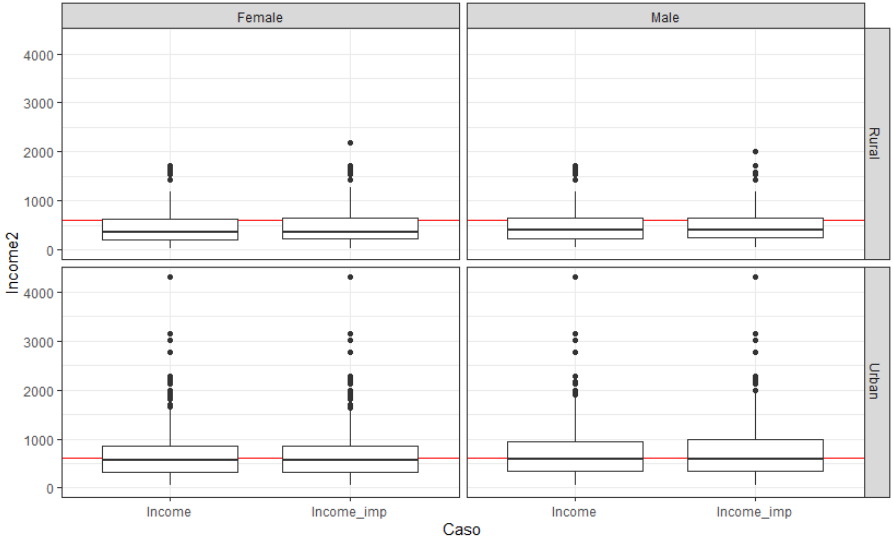


Figura 20: Imputación por el vecino más cercano con regresión

Introducción a la imputación múltiple.

Introducción a la imputación múltiple.

- Se crean múltiples copias del conjunto de datos.
- Los valores faltantes se imputan en cada copia usando modelos estadísticos.
- Análisis separados en cada copia generan resultados.
- Resultados combinados reflejan la incertidumbre causada por la imputación.

Ventajas de la Imputación Múltiple:

- Proporciona resultados más precisos y menos sesgados.
- Evita la pérdida de información al no eliminar observaciones con datos faltantes.
- Maneja la incertidumbre asociada con la imputación.

Introducción a la imputación múltiple.

Suponga que existe un conjunto de n datos que relaciona dos variables X, Y, a través del siguiente modelo de regresión simple:

$$y_i = \beta x_i + \varepsilon_i$$

Para todo individuo $i = 1, \ldots, n.$, de tal manera que los errores tienen distribución normal con $E(\varepsilon) = 0$ y $Var(\varepsilon) = \sigma^2$.

- Sea Y_{Obs} los valores observados para un conjunto de individuos de tamaño n_1.
- Sea Y_{NoObs} los valores **NO** observados de la variable Y de tamaño n_0, es decir, $n_1 + n_0 = n$.
- Suponga que sí fue posible observar los valores de la covariable X para todos los individuos en la muestra.

Simulación

- Simular un conjunto de datos con $n = 500$ observaciones.
- Pendiente de regresión (β) es 10, dispersión (σ) es 2.
- Introducir 200 valores faltantes en la variable respuesta Y.
- Uso de la función `rnorm` y `runif` en R para la simulación.

Introducción a la imputación múltiple.

El algoritmo de simulación.

```
generar <- function(n = 500, n_0 = 200,
                    beta = 10, sigma = 2){
  x <- runif(n)
  mu <- beta * x
  y <- mu + rnorm(n, mean = 0, sd = sigma)
  datos <- data.frame(x = x, y = y)
  faltantes <- sample(n, n_0)
  datos$faltantes <- "No"
  datos$faltantes[faltantes] <- "Si"
  datos$y.per <- y
  datos$y.per[faltantes] <- NA
  return(datos)
}
```

Introducción a la imputación múltiple.

```
set.seed(1234)
datos <- generar()
head(datos,12)
```

x	y	faltantes	y.per
0.1137	2.0109	No	2.011
0.6223	8.3432	No	8.343
0.6093	6.9971	No	6.997
0.6234	7.5602	Si	NA
0.8609	6.3364	No	6.336
0.6403	5.6621	No	5.662
0.0095	3.0489	No	3.049
0.2326	-0.1223	Si	NA
0.6661	7.1770	Si	NA
0.5143	5.9525	No	5.952
0.6936	8.8875	No	8.887
0.5450	4.7520	No	4.752

Introducción a la imputación múltiple.

```
library(patchwork)
p1 <- ggplot(data = datos, aes(x = x, y = y)) +
  geom_point() +
  geom_smooth(formula = y ~ x , method = "lm")

p2 <- ggplot(data = datos, aes(x = x, y = y.per)) +
  geom_point() +
  geom_smooth(formula = y ~ x , method = "lm")

p1 | p1
```

Introducción a la imputación múltiple.

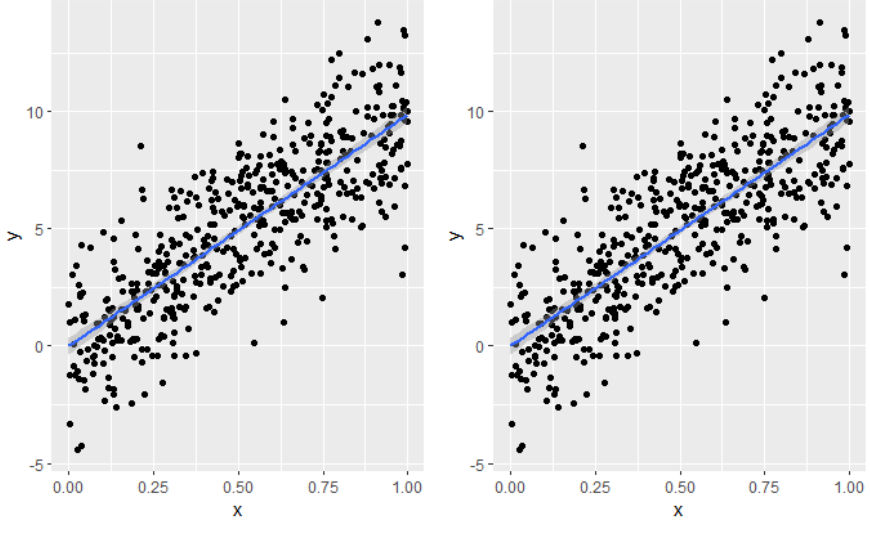


Figura 21: Imputación múltiple

Introducción a la imputación múltiple.

Ahora, dado el 40% de valores faltantes, es necesario imputar los datos faltantes. Para esto, utilizaremos la técnica de imputación múltiple propuesta por Rubin (1987)[1]. La idea consiste en generar $M > 1$ conjuntos de valores para los datos faltantes. Al final, el valor *imputado* corresponderá al promedio de esos M valores.

Hay varias maneras de realizar la imputación:

1. **Ingenua**: Esta clase de imputación carece de aleatoriedad y por tanto, la varianza de β va a ser subestimada.
2. **Bootstrap**: Se seleccionan m muestras bootstrap, y para cada una se estiman los parámetros β y σ para generar $\dot{y}_i$. Al final se promedian los m valores y se imputa el valor faltante.
3. **Bayesiana**: Se definen las distribuciones posteriores de β y σ para generar M valores de estos parámetros y por tanto M valores de $\dot{y}_i$. Al final se promedian los M valores y se imputa el valor faltante.

[1] Rubin, D. B. (1987). Multiple imputation for survey nonresponse.

Introducción a la imputación múltiple

Dado que el interés es la estimación de la pendiente de la regresión simple β, entonces la esperanza estimada al utilizar la metodología de imputación múltiple está dada por:

$$E(\widehat{\beta}|Y_{obs}) = E(E(\widehat{\beta}|Y_{obs}, Y_{mis})|Y_{obs})$$

Esta expresión es estimada por el promedio de las M estimaciones puntuales de $\widehat{\beta}$ sobre las M imputaciones, dado por:

$$\bar{\widehat{\beta}} = \frac{1}{M} \sum_{m=1}^{M} \widehat{\beta}_m$$

Introducción a la imputación múltiple

La varianza estimada al utilizar la metodología de imputación múltiple está dada por la siguiente expresión:

$$V(\hat{\beta}|Y_{obs}) = E(V(\hat{\beta}|Y_{obs}, Y_{mis})|Y_{obs}) + V(E(\hat{\beta}|Y_{obs}, Y_{mis})|Y_{obs})$$

La primera parte de la anterior expresión se estima como el promedio de las varianzas muestrales de $\hat{\beta}$ sobre las M imputaciones, dado por:

$$\bar{U} = \frac{1}{M} = \sum_{m=1}^{M} Var(\beta)$$

El segundo término se estima como la varianza muestral de las M estimaciones puntuales de $\hat{\beta}$ sobre las M imputaciones, dada por:

$$B = \frac{1}{M-1} = \sum_{m=1}^{M} (\hat{\beta}_m - \bar{\hat{\beta}})$$

Introducción a la imputación múltiple

Es necesario tener en cuenta un factor de corrección (puesto que M es finito). Por tanto, la estimación del segundo término viene dada por la siguiente expresión:

$$(1 + \frac{1}{M})B$$

Por tanto, la varianza estimada es igual a:

$$\hat{V}(\hat{\beta}|Y_{obs}) = \bar{U} + (1 + \frac{1}{M})B$$

Imputación Bootstrap

Una función que realiza esta imputación es la siguiente:

```
im.bootstrap <- function(datos, M = 15){
  library(dplyr)
  n <- nrow(datos)
  datos1 <- na.omit(datos)
  n1 <- nrow(datos1)
  n0 <- n - n1
  Ind <- is.na(datos$y.per)
  faltantes.boot <- NULL
  beta1 <- NULL
  sigma1 <- NULL
```

Continua...

Imputación Bootstrap

Continuando...

```
for (m in 1:M){
    datos.m <- dplyr::sample_n(datos1, n1, replace = TRUE)
    model1 <- lm(y ~ 0 + x, data = datos.m)
    beta <- model1$coeff
    sigma <- sqrt(anova(model1)[["Mean Sq"]][2])
    faltantes.boot <- rnorm(n0, datos$x[Ind] * beta,
                            sd = sigma)
    datos$y.per[Ind] <-  faltantes.boot
    model.input <- lm(y.per ~ 0 + x, data = datos)
    beta1[m] <- model.input$coeff
    sigma1[m] <- summary(model.input)$coeff[2]
  }
  beta.input <- mean(beta1)
  u.bar <- mean(sigma1 ^ 2)
  B <- var(beta1)
  beta.sd <- sqrt(u.bar + B + B/M)
  result <- list(new = datos, beta = beta.input,
                 sd = beta.sd)
}
```

Imputación Bootstrap

Al aplicar la función sobre el conjunto de datos creado, se obtienen las siguientes salidas:

```
set.seed(1234)
datos <- generar()
im.bootstrap(datos)$beta
```

```
[1] 9.784
```

```
im.bootstrap(datos)$sd
```

```
[1] 0.1947
```

```
head(im.bootstrap(datos)$new, 4)
```

x	y	faltantes	y.per
0.1137	2.011	No	2.011
0.6223	8.343	No	8.343
0.6093	6.997	No	6.997
0.6234	7.560	Si	4.709

Imputación Bootstrap

Nótese que existe una buena dispersión en los valores imputados.

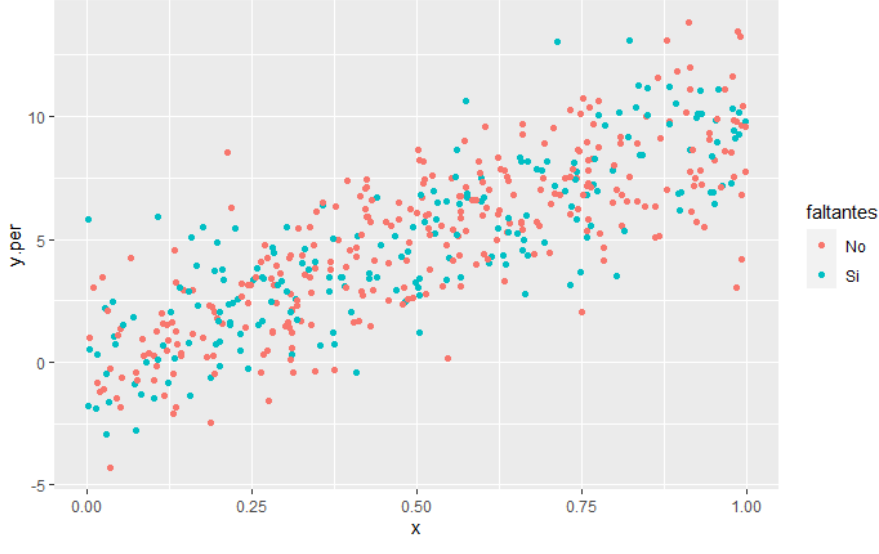


Figura 22: Regresión después de la imputando

Imputación Bootstrap en la encuesta.

Se ejemplificará la técnica de imputación múltiple para los datos de la encuesta

```
encuesta$Income_imp <- encuesta$Income_missin
encuesta$Employment_imp <- encuesta$Employment_missin
encuesta_obs <- filter(encuesta,
                       !is.na(Income_missin))
encuesta_no_obs <- filter(encuesta,
                          is.na(Income_missin))
n0 <- nrow(encuesta_no_obs)
n1 <- nrow(encuesta_obs)
```

Imputación Bootstrap en la encuesta.

```r
M = 10
set.seed(1234)
for (ii in 1:M) {
  vp <- paste0("Income_vp_", ii)
  vp2 <- paste0("Employment_vp_", ii)

  encuesta_temp <- encuesta_obs %>%
    sample_n(size = n1, replace = TRUE)

  mod <- lm(Income ~ Zone + Sex + Expenditure,
            data = encuesta_temp)
  mod.mult <- multinom(Employment ~ Zone + Sex + Expenditure,
                       data = encuesta_temp, )

  encuesta_no_obs[[vp]] <- predict(mod, encuesta_no_obs)
  encuesta_obs[[vp]] <- encuesta_obs$Income

  encuesta_no_obs[[vp2]] <- predict(mod.mult,
                                    encuesta_no_obs, type = "class")
  encuesta_obs[[vp2]] <- encuesta_obs$Employment
}
```

```
# weights:  15 (8 variable)
initial  value 1651.214270
```

Imputación Bootstrap en la encuesta.

Se seleccionan las variables de ingresos y sus 10 valores plausibles como se muestra a continuación:

```
select(encuesta_no_obs,
       Income, matches("Income_vp_"))[1:10,1:4]
```

Income	Income_vp_1	Income_vp_2	Income_vp_3
409.87	550.2	566.0	567.8
409.87	561.1	529.3	541.8
90.92	210.6	225.8	164.0
90.92	221.5	189.1	138.0
90.92	210.6	225.8	164.0
135.33	222.7	237.9	178.4
135.33	222.7	237.9	178.4
1539.75	784.9	801.1	846.8
336.00	507.9	472.8	439.7
685.48	593.0	558.1	540.9

Imputación Bootstrap en la encuesta.

```
encuesta <- bind_rows(encuesta_obs, encuesta_no_obs)
```

```
## Ordenando la base para gráfica
dat_plot10 <- tidyr::gather(
  encuesta %>% select(Zone,Sex,matches("Income_vp_")),
  key = "Caso", value = "Income2", -Zone,-Sex)

p1 <- ggplot(dat_plot10, aes(x = Income2, col = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
   theme(legend.position = "bottom") +
  geom_density(data = encuesta ,aes(x = Income),
               col =  "black", size = 1.2)

p1
```

Imputación por el vecino más cercano

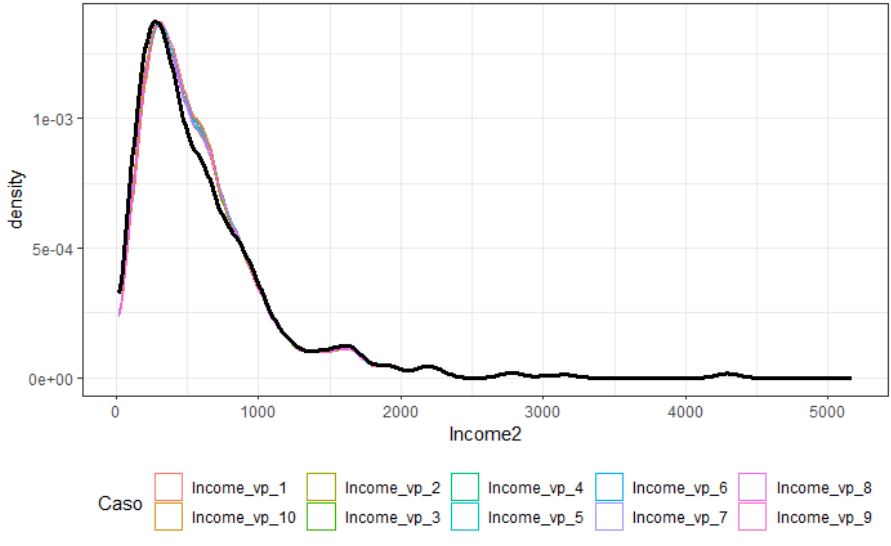


Figura 23: Densidad para los 10 valores plausible

Imputación Bootstrap en la encuesta.

```r
## Ordenando la base para gráfica
dat_plot11 <- tidyr::gather(
  encuesta %>%
  select(Zone,Sex, Employment,matches("Employment_vp_")),
  key = "Caso", value = "Employment2", -Zone,-Sex) %>%
  group_by(Caso,Employment2) %>% tally() %>%
  group_by(Caso) %>% mutate(prop = n/sum(n))

p1 <- ggplot(dat_plot11,
        aes(x = Employment2, y = prop,
            fill = Caso, color="red")) +
       geom_bar(stat="identity",
          position = position_dodge(width = 0.5))  +
   theme_bw() +
   theme(legend.position = "bottom") +
  scale_fill_manual(values = c("Employment" = "black"))
p1
```

Imputación por el vecino más cercano

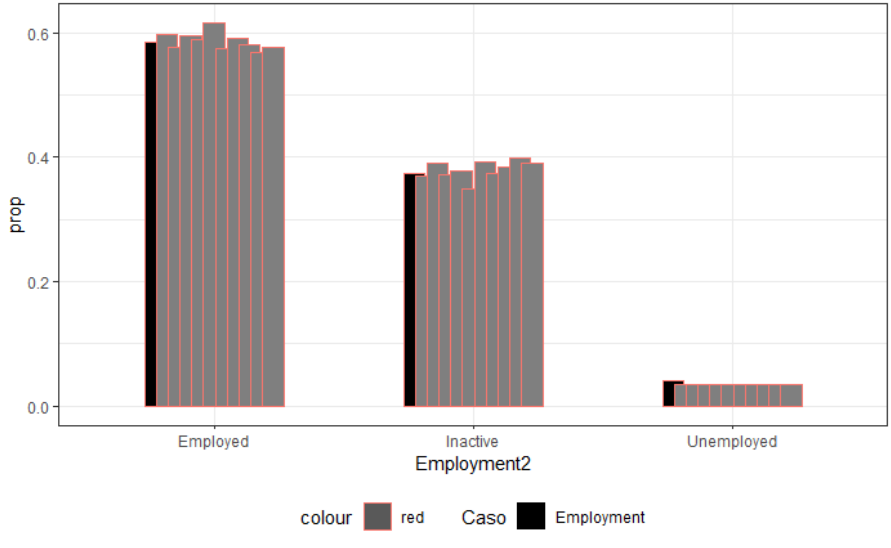


Figura 24: Regresión después de la imputando

Definir diseño de la muestra con `srvyr`

Se procede a definir el diseño muestral utilizado en este ejemplo y así poder hacer la estimación de los parámetros

```
library(srvyr)

diseno <- encuesta %>%
  as_survey_design(
    strata = Stratum,
    ids = PSU,
    weights = wk,
    nest = T
  )
```

Estimación del promedio con valores plausibles (vp)

Se estiman los ingresos medios para cada valor plausible junto con su varianza, como se muestra a continuación:

```
estimacion_vp <-  diseno %>%
 summarise(
   vp1 = survey_mean(Income_vp_1, vartype = c("var")),
   vp2 = survey_mean(Income_vp_2, vartype = c("var")),
   vp3 = survey_mean(Income_vp_3, vartype = c("var")),
   vp4 = survey_mean(Income_vp_4, vartype = c("var")),
   vp5 = survey_mean(Income_vp_5, vartype = c("var")),
   vp6 = survey_mean(Income_vp_6, vartype = c("var")),
   vp7 = survey_mean(Income_vp_7, vartype = c("var")),
   vp8 = survey_mean(Income_vp_8, vartype = c("var")),
   vp9 = survey_mean(Income_vp_9, vartype = c("var")),
   vp10 =survey_mean(Income_vp_10, vartype = c("var")))
```

Estimación del promedio con valores plausibles (vp)

vp	promedio	var
1	619.4	845.7
2	615.8	870.0
3	617.6	844.6
4	617.4	867.5
5	617.7	856.8
6	620.0	857.8
7	617.1	852.7
8	618.3	860.8
9	619.3	867.9
10	616.9	850.1

Estimación del promedio con valores plausibles (vp)

```
Media_vp = mean(estimacion_vp$promedio)
(Ubar = mean(estimacion_vp$var))
```

```
[1] 857.4
```

```
(B = var(estimacion_vp$promedio))
```

```
[1] 1.646
```

```
var_vp = Ubar + (1 + 1/M)
(resultado <- data.frame(Media_vp,
                         Media_vp_se = sqrt(var_vp)))
```

Media_vp	Media_vp_se
618	29.3

Estimación de la varianza con valores plausibles (vp)

otro parámetro de interés es la varianza de los ingresos.

```
estimacion_var_vp <-  diseno %>%
  summarise_at(vars(matches("Income_vp")),
               survey_var,  vartype = "var" )
```

vp	promedio	var
1	262690	3.075e+09
2	263092	3.080e+09
3	274370	3.238e+09
4	269127	3.165e+09
5	270450	3.165e+09
6	264993	3.107e+09
7	270916	3.176e+09
8	276070	3.252e+09
9	265061	3.111e+09
10	275462	3.258e+09

Estimación de la varianza con valores plausibles (vp)

Por último, se utilizan las ecuaciones mostradas anteriormente:

```
Media_var_vp <- mean(estimacion_var_vp$promedio)
(Ubar = mean(estimacion_var_vp$var))
```

```
[1] 3.163e+09
```

```
(B = var(estimacion_var_vp$promedio))
```

```
[1] 25796144
```

```
var_var_vp = Ubar + (1 + 1/M)*B
resultado$var_vp <- Media_var_vp
resultado$var_vp_se <- sqrt(var_var_vp)
```

Comparando resultados con valores plausibles (vp)

```
diseno %>% summarise(Media = survey_mean(Income),
                     Var = survey_var(Income))
```

Media	Media_se	Var	Var_se
607.6	31.71	282539	63156

```
resultado
```

Media_vp	Media_vp_se	var_vp	var_vp_se
618	29.3	269223	56489

Estimación de la proporción con valores plausibles (vp)

A continuación, se realizará la estimación de la proporción utilizando valores plausibles.

```
estimacion_prop_vp <-
  lapply(paste0("Employment_vp_",1:10),
       function(vp){
        diseno %>%
         group_by_at(vars(Employment = vp)) %>%
  summarise(prop = survey_mean(vartype = "var"),
            .groups = "drop") %>%
         mutate(vp = vp)
       }) %>% bind_rows()
```

Estimación de la varianza con valores plausibles (vp)

Se presenta la estimación de la proporción para cada uno de los 10 valores plausibles en cada categoría de la variable:

vp	Employment	prop	prop_var
1	Unemployed	0.0356	0e+00
1	Inactive	0.3754	2e-04
1	Employed	0.5891	2e-04
2	Unemployed	0.0356	0e+00
2	Inactive	0.3771	2e-04
2	Employed	0.5873	2e-04
3	Unemployed	0.0356	0e+00
3	Inactive	0.3868	2e-04
3	Employed	0.5776	2e-04
4	Unemployed	0.0356	0e+00
4	Inactive	0.3549	2e-04
4	Employed	0.6095	2e-04

Estimación de la varianza con valores plausibles (vp)

Por último, utilizando las ecuaciones de Rubin se obtiene la varianza estimada:

```
resultado = estimacion_prop_vp %>%
  group_by(Employment) %>%
  summarise(prop_pv = mean(prop),
            Ubar = mean(prop_var),
            B = var(prop)) %>%
  mutate(prop_pv_var = Ubar + (1 + 1/M)*B)
```

Comparando resultados con valores plausibles (vp)

```
diseno %>%  group_by(Employment ) %>%
 summarise(prop = survey_mean(vartype = "var"))
```

Employment	prop	prop_var
Unemployed	0.0429	1e-04
Inactive	0.3840	2e-04
Employed	0.5731	2e-04

```
resultado
```

Employment	prop_pv	Ubar	B	prop_pv_var
Unemployed	0.0356	0e+00	0e+00	0e+00
Inactive	0.3868	2e-04	2e-04	5e-04
Employed	0.5776	2e-04	2e-04	4e-04

Lectura de múltiples bases

Para realizar la lectura de múltiples bases debemos conocer las rutas donde estas estas guardadas para ello empleamos la función `file.list` del paquete base, que nos permite tener un listado completo de los archivos.

```
data_path <- list.files(
  "V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/",
  full.names = TRUE,
  pattern = "2020"
)
data_path <- data.frame(data_path) %>%
  mutate(pais = str_extract(data_path, "(?<=/)(\\w{3})(?=_)")) %>%
  filter(pais %in% c("CRI", "DOM", "SLV"))
data_path
```

Lectura de encuestas e imputación de datos

Dado que el proceso de imputación es un proceso más complejo que estimar promedios o proporciones se hace necesario construir una función adaptada a nuestras necesidades que nos ayude con el proceso

data_path	pais
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/CRI_2020N1.dta	CRI
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/DOM_2020N1.dta	DOM
V:/DAT/BADEHOG/BADEHOG_N/Estandarizadas/SLV_2020N.dta	SLV

Función para el proceso de imputación (Promedio sin condicionar)

Para el siguiente ejercicio se considero la variable ingresos (ingcorte) y se considera un valor perdido cuando `ingcorte = 0` que toma valores perdidos

```
imp_media <- function(input_file){
  ## Identificando el nombre del país
  pais = str_extract(input_file, "(?<=/)(\\w{3})(?=_)")
  ## Paso 1: lectura y selección de variables
  encuesta <- read_dta(input_file) %>%
    transmute(ingcorte,
              ingcorte_imp = ifelse(ingcorte==0,NA,ingcorte))
  ## Paso 2: Definir el método de imputación
  media = mean(encuesta$ingcorte_imp, na.rm = TRUE)
  ## Paso 3: Aplicar el método de imputación
  encuesta %<>%
    mutate(pais = pais,
      ingcorte_media = ifelse(is.na(ingcorte_imp),
                              media, ingcorte_imp))
  ## Paso 4: Retornar el resultado
  return(encuesta)
}
```

Procesando encuestas múltiples

Para aplicar la función `imp_media` en las diferentes encuestas ejecutamos la siguiente sintaxis.

```
library(furrr)
library(haven)
future::plan(multicore)
temp <- data_path %>%
    future_map_dfr(~imp_media(.x),.progress = FALSE)

temp %>% filter(is.na(ingcorte_imp)) %>% head( 10)
```

La función `future_map_dfr` es utilizada para trabajar con los elementos de una lista, ademas realizar procesamiento en paralelo, es decir, cada núcleo del ordenador opera una encuesta diferente, lo que permite reducir los tiempos de computacionales

Procesando encuestas múltiples (Resultado)

Los resultados se muestran en el orden de lectura de los archivos

ingcorte	ingcorte_imp	pais	ingcorte_media
0	NA	CRI	247150
0	NA	CRI	247150
0	NA	CRI	247150
0	NA	CRI	247150
0	NA	CRI	247150
0	NA	CRI	247150
0	NA	CRI	247150
0	NA	CRI	247150
0	NA	CRI	247150
0	NA	CRI	247150

Comparando resultados en los paises.

```
temp %>% group_by(pais) %>%
  summarise_all(mean,  na.rm = TRUE)
```

pais	ingcorte	ingcorte_imp	ingcorte_media
CRI	245640.0	247150.2	247150.2
DOM	10767.8	10784.5	10784.5
SLV	179.6	180.4	180.4

Comparando resultados en los paises.

```
temp %>% group_by(pais) %>%
  summarise_all(median,  na.rm = TRUE)
```

pais	ingcorte	ingcorte_imp	ingcorte_media
CRI	154756.7	155851.7	156625
DOM	8208.8	8224.0	8233
SLV	134.2	134.8	135

Función para el proceso de imputación (Promedio condicionado)

```
imp_media_grupo <- function(input_file){
  ## Identificando el nombre del país
  pais =  str_extract(input_file, "(?<=/)(\\w{3})(?=_)")
  ## Paso 1: lectura y selección de variables
  encuesta <- read_dta(input_file) %>%
    transmute(area_ee = haven::as_factor(area_ee),
              ingcorte,
              ingcorte_imp = ifelse(ingcorte==0,NA,ingcorte))
  ## Paso 2 y 3: Definir el método de imputación
  ## aplicar el método de imputación
  encuesta %<>% group_by(area_ee) %>%
    mutate(pais = pais,
      ingcorte_media = ifelse(is.na(ingcorte_imp),
                              mean(ingcorte_imp,na.rm = TRUE ),
                              ingcorte_imp))
  ## Paso 4: Retornar el resultado
  return(encuesta)
}
```

Procesando encuestas múltiples

Para aplicar la función `imp_media_grupo` en las diferentes encuestas ejecutamos la siguiente sintaxis.

```
future::plan(multicore)
temp <- data_path %>%
    future_map_dfr(~imp_media_grupo(.x),.progress = FALSE)
temp %>% filter(is.na(ingcorte_imp)) %>% head( 10) %>%
  data.frame()
```

Procesando encuestas múltiples (Resultado)

Los resultados se muestran en el orden de lectura de los archivos

area_ee	ingcorte	ingcorte_imp	pais	ingcorte_media
Urbana	0	NA	CRI	286937
Urbana	0	NA	CRI	286937
Urbana	0	NA	CRI	286937
Urbana	0	NA	CRI	286937
Urbana	0	NA	CRI	286937
Urbana	0	NA	CRI	286937
Urbana	0	NA	CRI	286937
Urbana	0	NA	CRI	286937
Urbana	0	NA	CRI	286937
Urbana	0	NA	CRI	286937

Comparando resultados en los paises.

```
temp %>% group_by(pais, area_ee) %>%
  summarise_all(mean,  na.rm = TRUE) %>%
  head(10) %>% data.frame()
```

pais	area_ee	ingcorte	ingcorte_imp	ingcorte_media
CRI	Urbana	285092.1	286936.9	286936.9
CRI	Rural	177415.0	178406.8	178406.8
DOM	Urbana	11358.1	11375.7	11375.7
DOM	Rural	9212.6	9227.2	9227.2
SLV	Urbana	223.0	223.9	223.9
SLV	Rural	133.8	134.5	134.5

Comparando resultados en los paises.

```
temp %>% group_by(pais,area_ee) %>%
  summarise_all(median,  na.rm = TRUE) %>%
  head(10) %>% data.frame()
```

pais	area_ee	ingcorte	ingcorte_imp	ingcorte_media
CRI	Urbana	183223.1	184653.3	186190.9
CRI	Rural	122450.0	123029.2	123774.4
DOM	Urbana	8546.7	8557.3	8563.3
DOM	Rural	7312.5	7328.8	7336.0
SLV	Urbana	166.7	167.2	167.7
SLV	Rural	104.7	105.0	105.4

Función para el proceso de imputación (Promedio condicionado)

```r
imp_media_lm <- function(input_file){
  ## Identificando el nombre del país
  pais =  str_extract(input_file, "(?<=/)(\\w{3})(?=_)")
  ## Paso 1: lectura y selección de variables
  encuesta <- read_dta(input_file) %>%
    transmute(area_ee,ingcorte, sexo = as.factor(sexo),edad,
              ingcorte_imp = ifelse(ingcorte==0,NA,ingcorte))
  ## Paso 2: Definir el método de imputación
  encuesta2 <- filter_all(encuesta,  all_vars(!is.na(.))) %>%
    select(-ingcorte)
  mod <- lm(ingcorte_imp~.,encuesta2)
  ## Paso 3: Aplicar el método de imputación
  encuesta$pred <- as.numeric(predict(mod,encuesta))
  encuesta %<>%
  mutate(pais = pais,ingcorte_lm = ifelse(is.na(ingcorte_imp),
                                pred, ingcorte_imp),
         pred = NULL) %>%
    select(pais,matches("ingcorte"))
  ## Paso 4: Retornar el resultado
  return(encuesta)
}
```

Procesando encuestas múltiples

Para aplicar la función `imp_media_lm` en las diferentes encuestas ejecutamos la siguiente sintaxis.

```
future::plan(multicore)
temp <- data_path %>%
    future_map_dfr(~imp_media_lm(.x),.progress = FALSE)
temp %>% filter(is.na(ingcorte_imp)) %>% head( 10)
```

Procesando encuestas múltiples (Resultado)

Los resultados se muestran en el orden de lectura de los archivos

pais	ingcorte	ingcorte_imp	ingcorte_lm
CRI	0	NA	258540
CRI	0	NA	254154
CRI	0	NA	327849
CRI	0	NA	278047
CRI	0	NA	293169
CRI	0	NA	308341
CRI	0	NA	291052
CRI	0	NA	284499
CRI	0	NA	232479
CRI	0	NA	210804

Comparando resultados en los paises.

```
temp %>% group_by(pais) %>%
  summarise_all(mean,  na.rm = TRUE)
```

pais	ingcorte	ingcorte_imp	ingcorte_lm
CRI	245640.0	247150.2	247125.5
DOM	10767.8	10784.5	10784.5
SLV	179.6	180.4	180.4

Comparando resultados en los paises.

```
temp %>% group_by(pais) %>%
  summarise_all(median,  na.rm = TRUE)
```

pais	ingcorte	ingcorte_imp	ingcorte_lm
CRI	154756.7	155851.7	156528
DOM	8208.8	8224.0	8229
SLV	134.2	134.8	135

¡Gracias!

Email: andres.gutierrez@cepal.org