Análisis de encuestas de hogares con R

Modulo 8: Métodos de imputación

CEPAL - Unidad de Estadísticas Sociales

Tabla de contenidos I

Introducción

Introducción valores perdidos

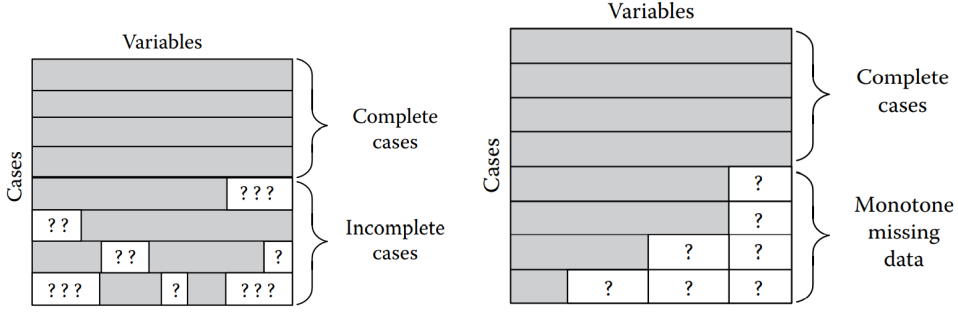
- Sea $X_{n\times p} = x_{ij}$ una matriz completa (sin valores perdidos), de tal forma que X_{ij} es el valor de la variable j, $j = 1, \dots, p$ en el caso i, $i = 1, \dots, n$.
- Sea $M_{n\times p} = m_{ij}$, donde $m_{ij} = 1$ si x_{ij} es un dato perdido y $m_{ij} = 0$ si x_{ij} está presente.
- 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("Imagenes/06_MLG1/ENIGH_HND_Hogar.rds")
encuesta <- encuesta %>% # Base de datos.
  transmute(
    LLAVE_HOGAR,
    F1_A0_UPM,
    estrato = haven::as_factor(F1_A0_ESTRATO),
    dam = haven::as_factor(F1_A0_DEPARTAMENTO),
    Area = haven::as_factor(F1_A0_AREA),
    ingreso_per  = ifelse(YDISPONIBLE_PER < 0 , 0 , YDISPONIBLE_PER) ,
    gasto_per = GASTO_CORRIENTE_HOGAR / CANTIDAD_PERSONAS,
    pobreza_LP = case_when(
      ingreso_per < 3046 & Area == "1. Urbana" ~ "1",
      ingreso_per < 1688 & Area == "2. Rural" ~ "1",
      TRUE ~ "0"
    ),
     TIPOVIVIENDA = haven::as_factor(F1_A1_P1_TIPOVIVIENDA),
     TIENEVEHICULOS = haven::as_factor(F2_A2_P1_TIENEVEHICULOS),
     TECHOVIVIENDA = haven::as_factor(F1_A1_P4_TECHOVIVIENDA),
     log_ingreso_per = log(ingreso_per  + 500),
     log_gasto_per = log(gasto_per  + 500),
    Factor
  )

(tab_antes <- prop.table(table(encuesta$pobreza_LP)))
```

0	1
0.8152	0.1848

Creando valores perdidos

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

```
set.seed(1234)
encuesta_MCAR <-  sample_frac(encuesta, 0.7 )
dat_plot <- bind_rows(
  list(encuesta_MCAR = encuesta_MCAR,
       encuesta = encuesta), .id = "Caso"  )
```

Creando valores perdidos

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

p2 <- ggplot(dat_plot, aes(x=TIPOVIVIENDA, y = log_ingreso_per)) +
  geom_boxplot() + facet_grid(.~Caso) +theme_bw()+
  geom_hline(yintercept = mean(encuesta$log_ingreso_per),
             col = "red")
library(patchwork)
p0 <- p1|p2
```

Creando valores perdidos

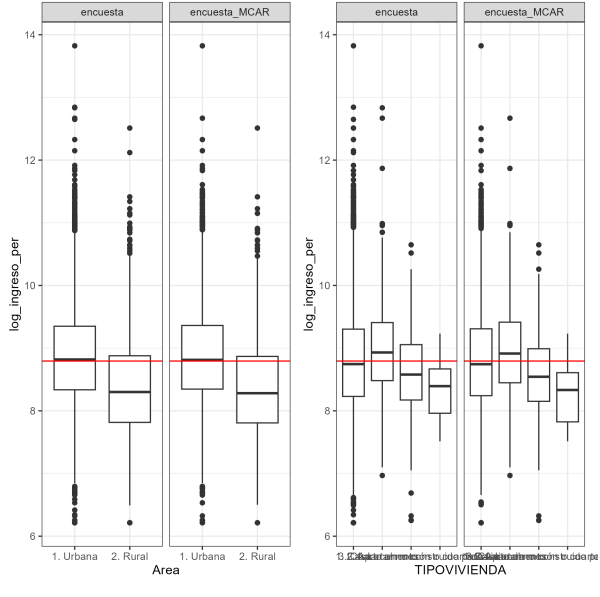


Figura 1: Valores perdidos con el esquema MCAR

Creando valores perdidos

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

p2 <- ggplot(dat_plot, aes(x = log_ingreso_per, fill = Caso)) +
  geom_density(alpha = 0.3) + facet_grid(.~TIPOVIVIENDA) +
  theme_bw()+
  geom_vline(xintercept = mean(encuesta$log_ingreso_per),
             col = "red") +
  theme(legend.position = "none")
p0 <- (p1/p2)
```

Creando valores perdidos

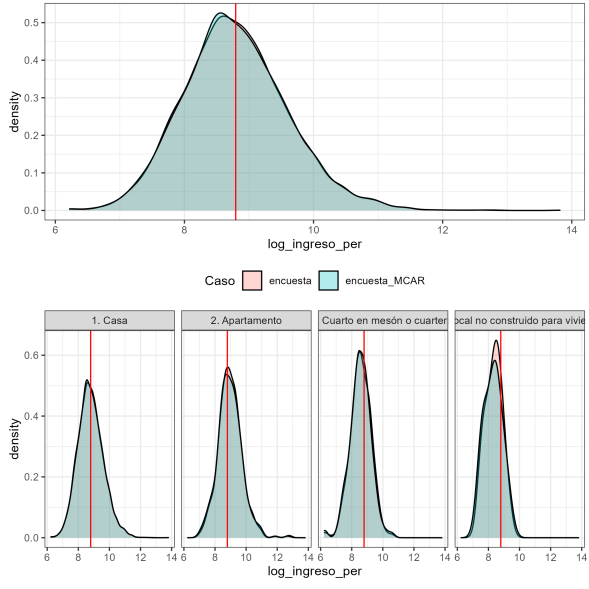


Figura 2: Densidades del ingreso con Valores perdidos por el esquema MCAR

Creando valores perdidos

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

p2 <- ggplot(dat_plot, aes(x = log_gasto_per, fill = Caso)) +
  geom_density(alpha = 0.3) + facet_grid(.~TIPOVIVIENDA) +
  theme_bw()+
  geom_vline(xintercept = mean(encuesta$log_gasto_per),
             col = "red") +
  theme(legend.position = "none")
p0 <- (p1/p2)
```

Creando valores perdidos

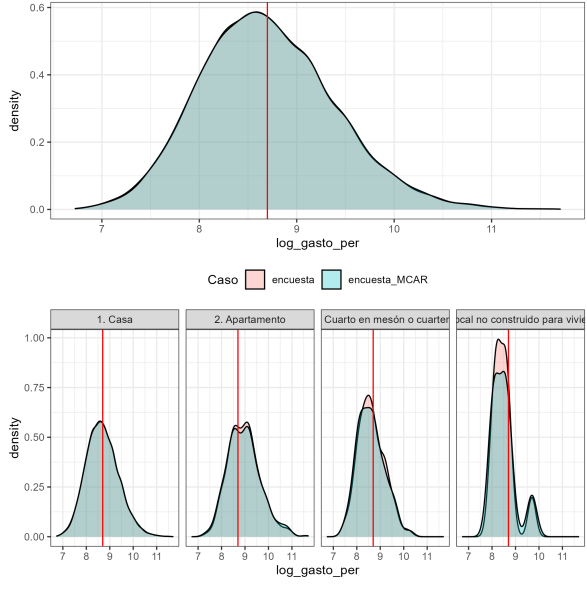


Figura 3: Densidades del gasto con valores perdidos por el esquema MCAR

Creando valores perdidos

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

Creando valores perdidos

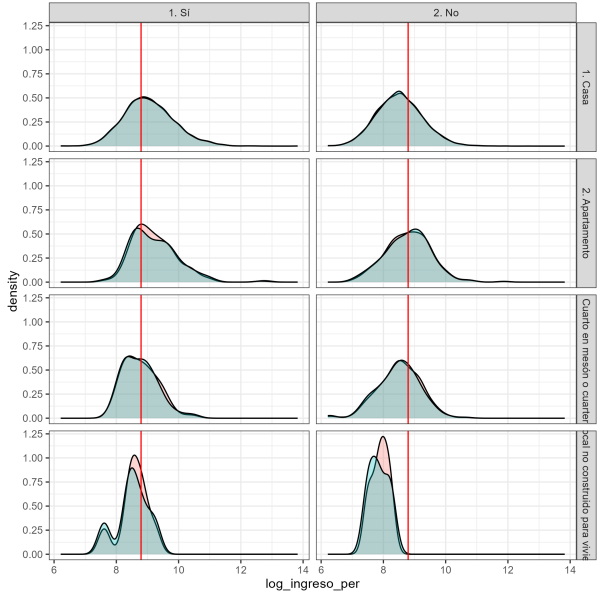


Figura 4: Densidades del ingreso con Valores perdidos por el esquema MCAR

Creando valores perdidos

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

Creando valores perdidos

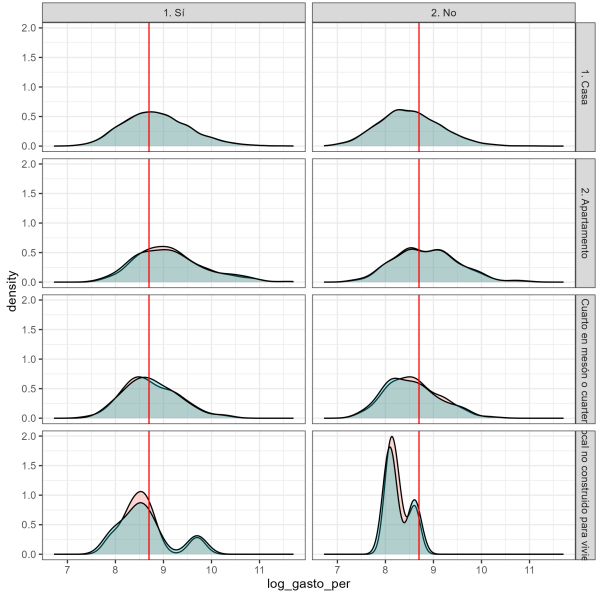


Figura 5: Densidades del ingreso con valores perdidos por el esquema MCAR

Creando valores perdidos

simulemos ahora una pérdida de información MAR como sigue:

```
library(TeachingSampling)
temp_estrato <- paste0(encuesta$dam, encuesta$Area)
temp_Nh <- as.data.frame(table(temp_estrato)) %>%
  rename(Nh = Freq) %>%
  mutate(nh = ceiling(Nh*.1))
dim(temp_Nh)
```

```
[1] 36  3
```

Creando valores perdidos

temp_estrato	Nh	nh
1. Atlántida1. Urbana	562	57
1. Atlántida2. Rural	49	5
10. Intibuca1. Urbana	127	13
10. Intibuca2. Rural	36	4
11. Islas De La Bahía1. Urbana	184	19
11. Islas De La Bahía2. Rural	93	10
12. La Paz1. Urbana	104	11
12. La Paz2. Rural	64	7
13. Lempira1. Urbana	81	9
13. Lempira2. Rural	50	5
14. Ocotepeque1. Urbana	125	13
14. Ocotepeque2. Rural	12	2
15. Olancho1. Urbana	630	63
15. Olancho2. Rural	106	11
16. Santa Bárbara1. Urbana	226	23
16. Santa Bárbara2. Rural	96	10

Creando valores perdidos

```
set.seed(1234)
sel <- S.STSI(S = temp_estrato,
              Nh = temp_Nh$Nh,
              nh = temp_Nh$nh)
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 = log_ingreso_per)) +
   geom_hline(yintercept = mean(encuesta$log_ingreso_per),
              col = "red") +
  geom_boxplot() +
  facet_grid(Area ~TIPOVIVIENDA) + theme_bw()
```

Creando valores perdidos

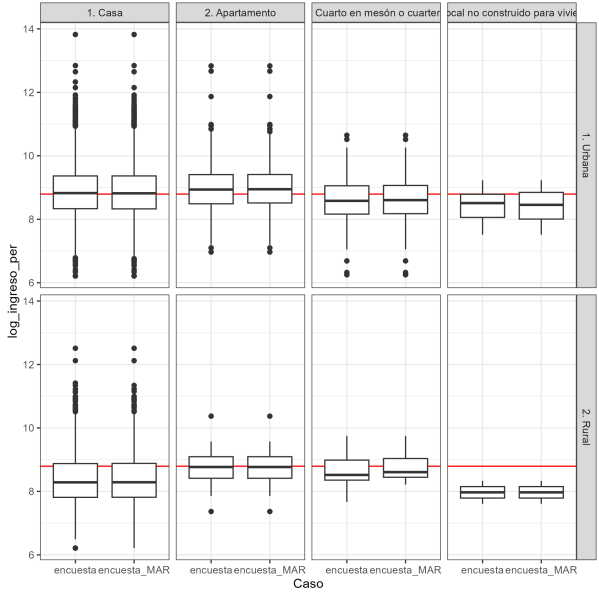


Figura 6: Valores perdidos con el esquema MCAR para el ingreso

Creando valores perdidos

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

Creando valores perdidos

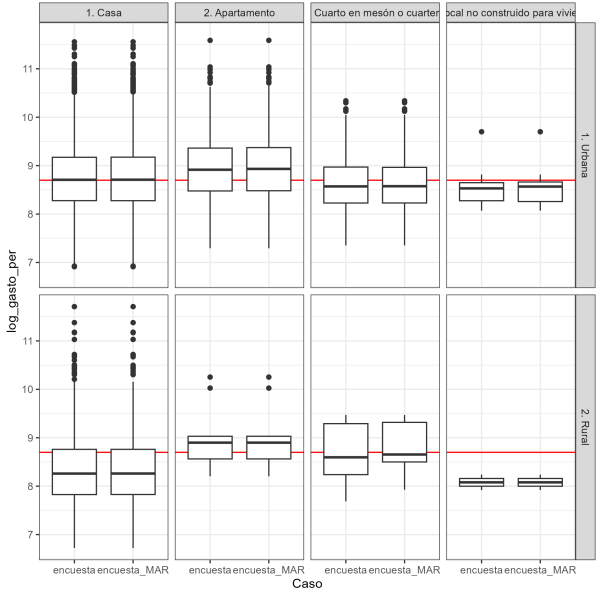


Figura 7: Valores perdidos con el esquema MCAR para el gasto

Creando valores perdidos

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

p2 <- ggplot(dat_plot2, aes(x = log_ingreso_per, fill = Caso)) +
  facet_grid(. ~ TIPOVIVIENDA) +
  geom_density(alpha = 0.3) + theme_bw() +
  theme(legend.position = "none") +
  geom_vline(xintercept = mean(encuesta$log_ingreso_per),
             col = "red")
p0 <- p1 / p2
```

Creando valores perdidos

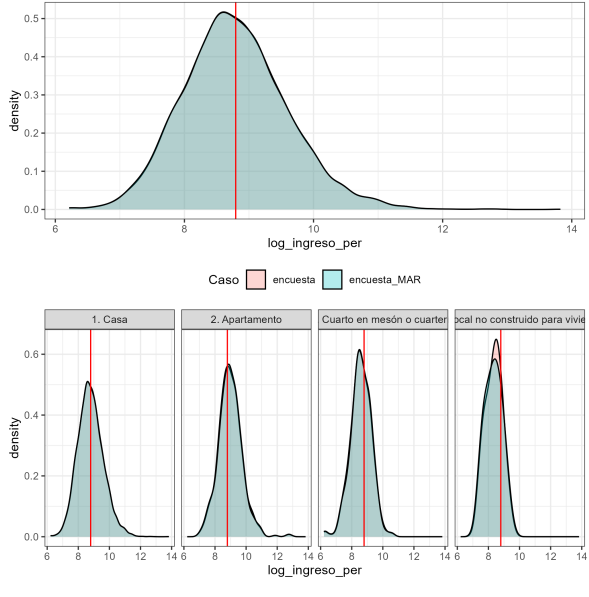


Figura 8: Densidades del ingreso con valores perdidos por el esquema MCAR

Creando valores perdidos

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

p2 <- ggplot(dat_plot2,
             aes(x = log_gasto_per, fill = Caso)) +
  facet_grid(.~TIPOVIVIENDA) +
  geom_density(alpha = 0.3) + theme_bw() +
  theme(legend.position = "none") +
  geom_vline(
    xintercept = mean(encuesta$log_gasto_per),
             col = "red")
p0 <- p1/p2
```

Creando valores perdidos

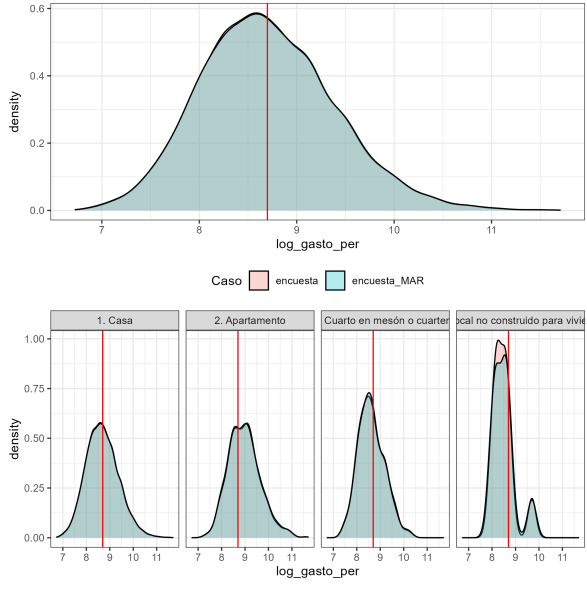


Figura 9: Densidades del gastos con valores perdidos por el esquema MCAR

Creando valores perdidos

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

```
n <- ceiling(nrow(encuesta) * .80)
encuesta_MNAR <- encuesta %>%
  arrange((log_ingreso_per)) %>%
  slice(1:n)

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

Creando valores perdidos

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

Creando valores perdidos

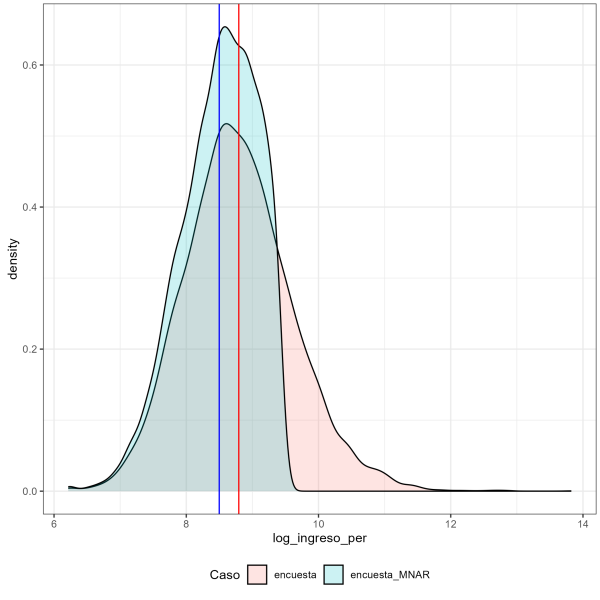


Figura 10: Densidades del ingreso con valores perdidos por el esquema MNAR

Creando valores perdidos

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

Creando valores perdidos

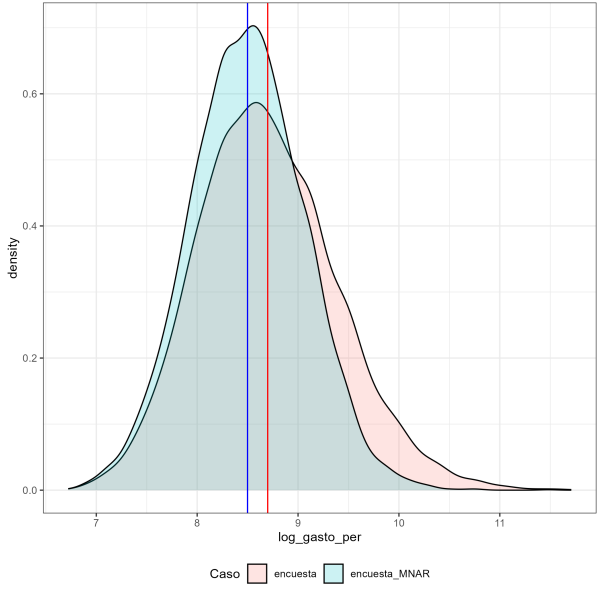


Figura 11: Densidades del gasto con valores perdidos por el esquema MNAR

Creando valores perdidos

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

Creando valores perdidos

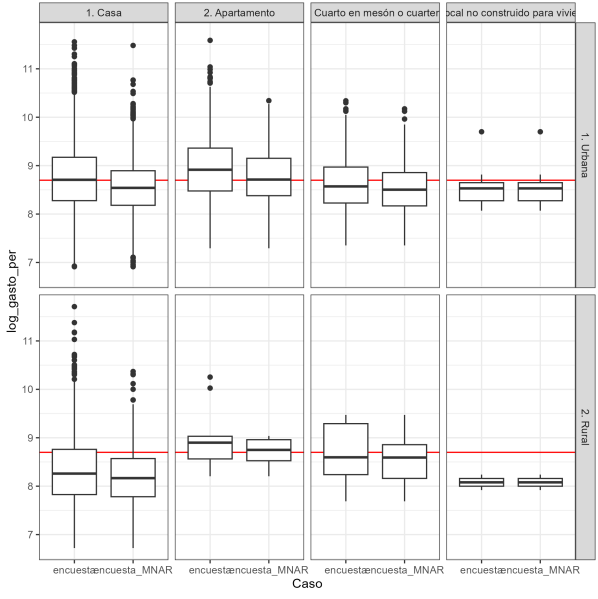


Figura 12: Impacto de la no respuesta con un esquema NMAR

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 %>%
    transmute(
    LLAVE_HOGAR,
    pobreza_LP_missin = pobreza_LP ,
    TIENEVEHICULOS_missin = TIENEVEHICULOS ,
    TECHOVIVIENDA_missin = TECHOVIVIENDA,
    log_ingreso_per_missin = log_ingreso_per ,
    log_gasto_per_missin =   log_gasto_per
    )
)
```

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(Area) %>%
  summarise(
    log_gasto_per = sum(is.na(log_gasto_per_missin) / n()))
```

Area	log_gasto_per
1. Urbana	0.2995
2. Rural	0.3030

```
encuesta %>% group_by(TIENEVEHICULOS) %>%
summarise(
    log_gasto_per = sum(is.na(log_gasto_per_missin) / n()))
```

TIENEVEHICULOS	log_gasto_per
1. Sí	0.2908
2. No	0.3089

Imputación de valores perdidos.

```
encuesta %>% group_by(TIPOVIVIENDA) %>%
summarise(
    log_gasto_per = sum(is.na(log_gasto_per_missin) / n()))
```

TIPOVIVIENDA	log_gasto_per
1. Casa	0.2986
2. Apartamento	0.3074
3. Cuarto en mesón o cuartería	0.3183
4. Local no construido para vivienda	0.2500

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 log_gasto_per_missin y se almacena en una nueva variable llamada log_gasto_per_imp

```
promedio_ingreso <- mean(encuesta$log_ingreso_per_missin, na.rm = TRUE)
promedio_gasto <- mean(encuesta$log_gasto_per_missin, na.rm = TRUE)
encuesta %<>%
  mutate(
    log_ingreso_per_imp = ifelse(is.na(log_ingreso_per_missin),
                                 promedio_ingreso, log_ingreso_per_missin),
    log_gasto_per_imp = ifelse(is.na(log_gasto_per_missin),
                               promedio_gasto, log_gasto_per_missin))
sum(is.na(encuesta$log_ingreso_per_imp))
```

```
[1] 0
```

```
sum(is.na(encuesta$log_gasto_per_imp))
```

```
[1] 0
```

Imputación por la media no condicional.

```
## Ordenando la base para gráfica
dat_plot4 <- tidyr::gather(
  encuesta %>% dplyr::select(Area,TIPOVIVIENDA,log_ingreso_per, log_ingreso
  key = "Caso", value = "log_ingreso_per2", -Area,-TIPOVIVIENDA)

p1 <- ggplot(dat_plot4, aes(x = log_ingreso_per2, fill = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
   theme(legend.position = "bottom") +
  geom_vline(
    xintercept = mean(encuesta$log_ingreso_per),
               col = "red") +
  geom_vline(
    xintercept = mean(encuesta$log_ingreso_per_imp),
               col = "blue")
p1
```

Imputación por la media no condicional.

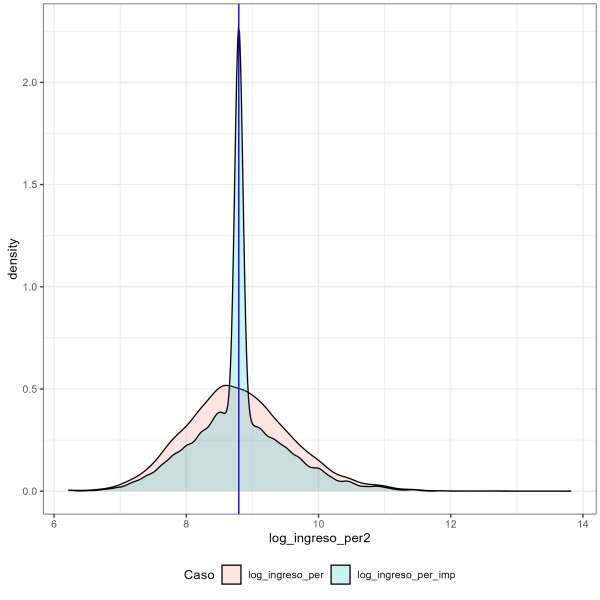


Figura 13: Imputación por la media no condicional para el ingreso

Imputación por la media no condicional.

```
## Ordenando la base para gráfica
dat_plot4 <- tidyr::gather(
  encuesta %>% dplyr::select(Area,TIPOVIVIENDA,log_gasto_per,
                             log_gasto_per_imp),
  key = "Caso", value = "log_gasto_per2", -Area,-TIPOVIVIENDA)

p1 <- ggplot(dat_plot4, aes(x = log_gasto_per2, fill = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
   theme(legend.position = "bottom") +
  geom_vline(
    xintercept = mean(encuesta$log_gasto_per),
              col = "red") +
  geom_vline(
    xintercept = mean(encuesta$log_gasto_per_imp),
              col = "blue")
p1
```

Imputación por la media no condicional.

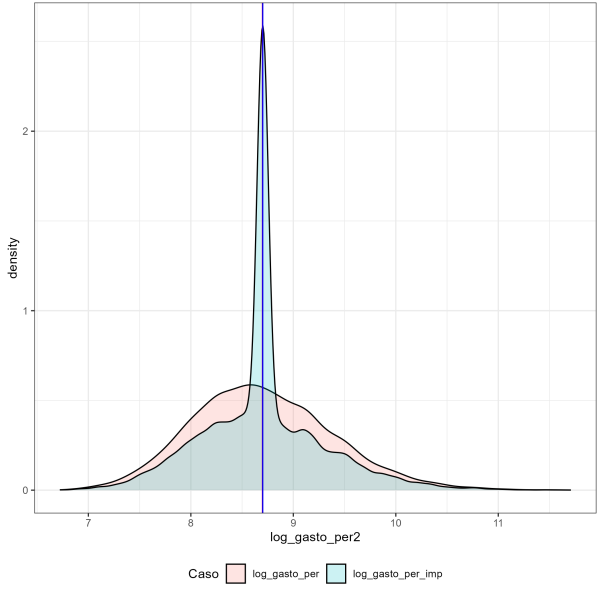


Figura 14: Imputación por la media no condicional para el gasto

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(estrato) %>%
  mutate(
    log_gasto_per_imp = ifelse(is.na(log_gasto_per_missin),
     mean(log_gasto_per_missin, na.rm = TRUE),
     log_gasto_per_missin)) %>% data.frame()

sum(is.na(encuesta$log_gasto_per_imp))
```

```
[1] 0
```

```
encuesta %<>% group_by(estrato) %>%
  mutate(
    log_ingreso_per_imp = ifelse(is.na(log_ingreso_per_missin),
     mean(log_ingreso_per_missin, na.rm = TRUE),
     log_ingreso_per_missin)) %>% data.frame()
sum(is.na(encuesta$log_ingreso_per_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(
  log_ingreso = mean(log_ingreso_per),
  log_ingreso_sd = sd(log_ingreso_per),
  log_ingreso_imp_ = mean(log_ingreso_per_imp),
  log_ingreso_imp_sd = sd(log_ingreso_per_imp))
```

log_ingreso	log_ingreso_sd	log_ingreso_imp_	log_ingreso_imp_sd
8.794	0.8157	8.796	0.7045

Imputación por la media condicional.

```
encuesta %>% summarise(
  log_gasto = mean(log_gasto_per),
  log_gasto_sd = sd(log_gasto_per),
  log_gasto_imp_ = mean(log_gasto_per_imp),
  log_gasto_imp_sd = sd(log_gasto_per_imp))
```

log_gasto	log_gasto_sd	log_gasto_imp_	log_gasto_imp_sd
8.699	0.6876	8.703	0.5978

Imputación por la media condicional.

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

$$BR = \frac{ingreso - ingreso_{imp}}{ingreso} \times 100\%$$

```
100*(8.794  - 8.797)/8.794
```

```
[1] -0.03411
```

Imputación por la media condicional.

```
encuesta %>%group_by(Area) %>%  summarise(
  log_gasto = mean(log_gasto_per),
  log_gasto_sd = sd(log_gasto_per),
  log_gasto_imp_ = mean(log_gasto_per_imp),
  log_gasto_imp_sd = sd(log_gasto_per_imp)) %>%
  mutate(BR = 100*(log_gasto - log_gasto_imp_)/log_gasto )
```

Area	log_gasto	log_gasto_sd	log_gasto_imp_	log_gasto_imp_sd	BR
1. Urbana	8.762	0.6632	8.767	0.5704	-0.0575
2. Rural	8.333	0.7127	8.333	0.6181	0.0024

Imputación por la media condicional.

```
encuesta %>% group_by(TIPOVIVIENDA) %>%  summarise(
  log_gasto = mean(log_gasto_per),
  log_gasto_sd = sd(log_gasto_per),
  log_gasto_imp = mean(log_gasto_per_imp),
  log_gasto_imp_sd = sd(log_gasto_per_imp)) %>%
  mutate(BR = 100*(log_gasto - log_gasto_imp)/log_gasto )
```

TIPOVIVIENDA	log_gasto	log_gasto_sd	log_gasto_imp	log_gasto_imp_sd	BR
1. Casa	8.684	0.6904	8.690	0.5993	-0.0598
2. Apartamento	8.966	0.6777	8.930	0.6014	0.4075
3. Cuarto en mesón o cuartería	8.623	0.5615	8.662	0.4943	-0.4604
4. Local no construido para vivienda	8.481	0.4675	8.586	0.4944	-1.2340

Imputación por la media condicional.

```
## Ordenando la base para gráfica
dat_plot5 <- tidyr::gather(
  encuesta %>% dplyr::select(Area,TIPOVIVIENDA,log_gasto_per,
                             log_gasto_per_imp),
  key = "Caso", value = "log_gasto_per2", -Area,-TIPOVIVIENDA)

p1 <- ggplot(dat_plot5, aes(x = log_gasto_per2, fill = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
   theme(legend.position = "bottom") +
  geom_vline(
    xintercept = mean(encuesta$log_gasto_per),
              col = "red") +
  geom_vline(
    xintercept = mean(encuesta$log_gasto_per_imp),
              col = "blue")
p1
```

Imputación por la media condicional.

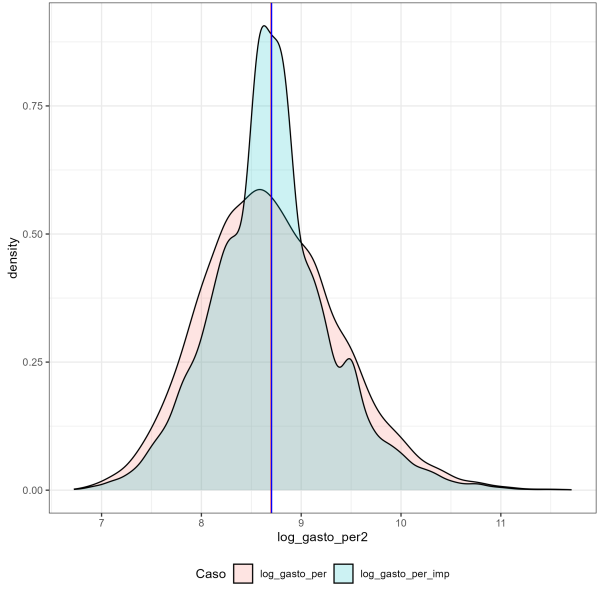


Figura 15: Imputación por la media no condicional sobre el gasto

Imputación por la media condicional.

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

```
p1 <- ggplot(dat_plot5, aes(x= Caso, y = log_gasto_per2)) +
   geom_hline(yintercept = mean(encuesta$log_gasto_per),
              col = "red") +  geom_boxplot() +
  facet_grid(Area~TIPOVIVIENDA) + theme_bw()
p1
```

Imputación por la media condicional.

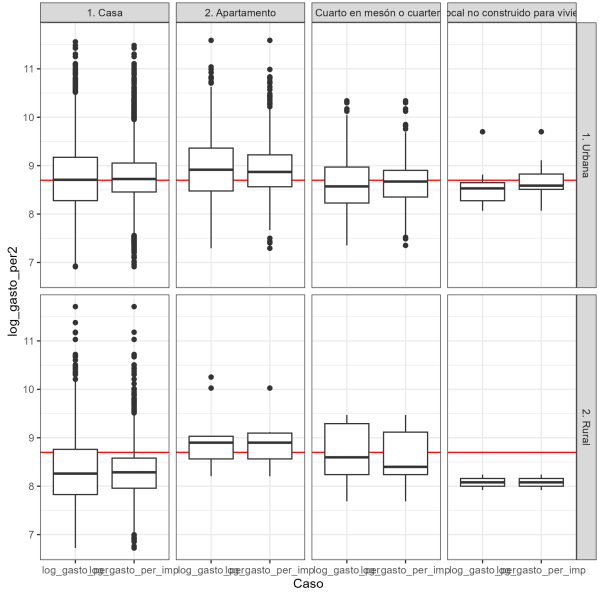


Figura 16: Imputación por la media no condicional TIPOVIVIENDAo y Área

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$log_gasto_per_missin))
receptor <- which(is.na(encuesta$log_gasto_per_missin))
encuesta$log_gasto_per_imp <- encuesta$log_gasto_per_missin
set.seed(1234)
for(ii in receptor){
don_ii <- sample(x = donante, size = 1)
encuesta$log_gasto_per_imp[ii] <-
  encuesta$log_gasto_per_missin[don_ii]
}
sum(is.na(encuesta$log_gasto_per_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(
  log_gasto = mean(log_gasto_per),
  log_gasto_sd = sd(log_gasto_per),
  log_gasto_imp_ = mean(log_gasto_per_imp),
  log_gasto_imp_sd = sd(log_gasto_per_imp)) %>%
  mutate(BR = 100*(log_gasto - log_gasto_imp_)/log_gasto )
```

log_gasto	log_gasto_sd	log_gasto_imp_	log_gasto_imp_sd	BR
8.699	0.6876	8.698	0.6855	0.0062

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(Area) %>%  summarise(
  log_gasto = mean(log_gasto_per),
  log_gasto_sd = sd(log_gasto_per),
  log_gasto_imp_ = mean(log_gasto_per_imp),
  log_gasto_imp_sd = sd(log_gasto_per_imp))%>%
  mutate(BR = 100*(log_gasto - log_gasto_imp_)/log_gasto )
```

Area	log_gasto	log_gasto_sd	log_gasto_imp_	log_gasto_imp_sd	BR
1. Urbana	8.762	0.6632	8.745	0.6690	0.1991
2. Rural	8.333	0.7127	8.431	0.7184	-1.1722

Imputación por hot-deck

```
encuesta %>% group_by(TIPOVIVIENDA) %>%   summarise(
  log_gasto = mean(log_gasto_per),
  log_gasto_sd = sd(log_gasto_per),
  log_gasto_imp_ = mean(log_gasto_per_imp),
  log_gasto_imp_sd = sd(log_gasto_per_imp)
) %>%
  mutate(BR = 100 * (log_gasto - log_gasto_imp_) / log_gasto)
```

TIPOVIVIENDA	log_gasto	log_gasto_sd	log_gasto_imp_	log_gasto_imp_sd	BR
1. Casa	8.684	0.6904	8.690	0.6850	-0.0692
2. Apartamento	8.966	0.6777	8.855	0.7202	1.2409
3. Cuarto en mesón o cuartería	8.623	0.5615	8.642	0.6171	-0.2238
4. Local no construido para vivienda	8.481	0.4675	8.675	0.7111	-2.2922

Imputación por hot-deck

```
## Ordenando la base para gráfica
dat_plot6 <- tidyr::gather(
  encuesta %>% dplyr::select(Area,TIPOVIVIENDA,log_gasto_per,
                             log_gasto_per_imp),
  key = "Caso", value = "log_gasto_per2", -Area,-TIPOVIVIENDA)

p1 <- ggplot(dat_plot6, aes(x = log_gasto_per2, fill = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
   theme(legend.position = "bottom") +
  geom_vline(
    xintercept = mean(encuesta$log_gasto_per),
              col = "red") +
  geom_vline(
    xintercept = mean(encuesta$log_gasto_per_imp),
              col = "blue")
p1
```

Imputación por hot-deck

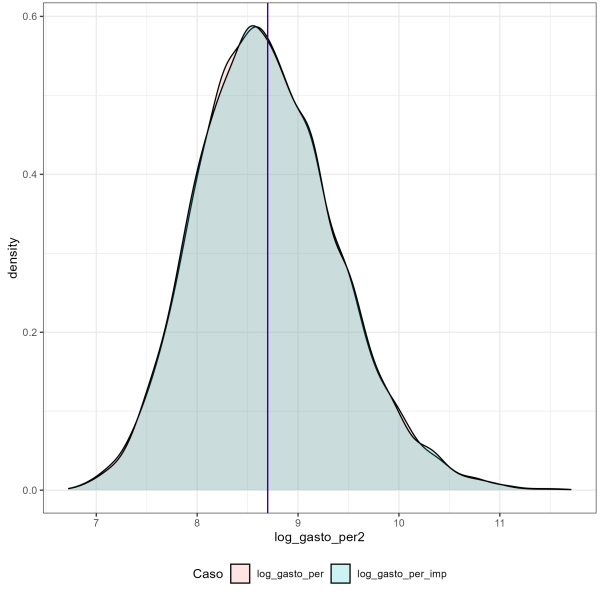


Figura 17: Imputación por hot-deck

Imputación por hot-deck

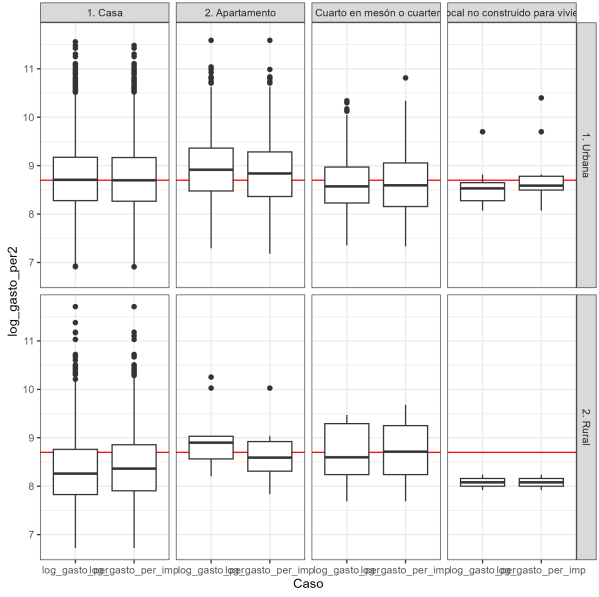


Figura 18: Imputación por hot-deck por TIPOVIVIENDA y área

Imputación por hot-deck

Se implementa el método de imputación pero para la variable tiene vehículo

```
donante <- which(!is.na(encuesta$log_gasto_per_missin))
receptor <- which(is.na(encuesta$log_gasto_per_missin))
encuesta$TIENEVEHICULOS_imp <- encuesta$TIENEVEHICULOS_missin
(prop <- prop.table(
  table(na.omit(encuesta$TIENEVEHICULOS_imp))))
```

1. Sí	2. No
0.4967	0.5033

Imputación por hot-deck estado de ocupación

```
set.seed(1234)
imp <- sample(size = length(receptor),
  c("1. Sí", "2. No"),
       prob = prop, replace = TRUE     )
encuesta$TIENEVEHICULOS_imp[receptor] <- imp
sum(is.na(encuesta$TIENEVEHICULOS_imp))
```

```
[1] 0
```

Imputación por hot-deck

Resultados antes de la imputación

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

1. Sí	2. No	NA
0.3477	0.3523	0.3

Resultados después de la imputación

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

1. Sí	2. No	NA
0.4976	0.5024	0

Imputación por hot-deck

Resultados antes de la imputación

TECHOVIVIENDA	1. Sí	2. No	Sum	
1. Urbana	0.3014	0.2962	0.8532	0.2555
2. Rural	0.0463	0.0560	0.1468	0.0445
Sum	0.3477	0.3523	1.0000	0.3000
NA	0.0000	0.0000	0.0000	0.0000

Resultados después de la imputación

TECHOVIVIENDA	1. Sí	2. No	Sum	
1. Urbana	0.4296	0.4236	0.8532	0
2. Rural	0.0680	0.0788	0.1468	0
Sum	0.4976	0.5024	1.0000	0
NA	0.0000	0.0000	0.0000	0

Imputación por hot-deck

Resultados antes de la imputación

TIPOVIVIENDA	1. Sí	2. No	Sum	
1. Casa	0.3216	0.3024	0.8898	0.2657
2. Apartamento	0.0173	0.0265	0.0632	0.0194
3. Cuarto en mesón o cuartería	0.0081	0.0230	0.0456	0.0145
4. Local no construido para vivienda	0.0007	0.0003	0.0014	0.0003
5. Otro, especifique	0.0000	0.0000	0.0000	0.0000
Sum	0.3477	0.3523	1.0000	0.3000
NA	0.0000	0.0000	0.0000	0.0000

Resultados después de la imputación

TIPOVIVIENDA	1. Sí	2. No	Sum	
1. Casa	0.4545	0.4353	0.8898	0
2. Apartamento	0.0270	0.0362	0.0632	0
3. Cuarto en mesón o cuartería	0.0152	0.0304	0.0456	0
4. Local no construido para vivienda	0.0009	0.0005	0.0014	0
5. Otro, especifique	0.0000	0.0000	0.0000	0
Sum	0.4976	0.5024	1.0000	0
NA	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 tipo de vivienda utilizando modelos de regresión lineal múltiple y multinomial, respectivamente, con covariables como área, TIPOVIVIENDA y empleo.

```
require(nnet)
encuesta$log_gasto_per_imp <- encuesta$log_gasto_per_missin
encuesta$TECHOVIVIENDA_imp <- encuesta$TECHOVIVIENDA_missin
encuesta_obs <- filter(encuesta,
                       !is.na(log_gasto_per_missin))
encuesta_no_obs <- filter(encuesta,
                          is.na(log_gasto_per_missin))
```

Imputación por regresión

Modelo para imputación del ingreso

```
mod <- lm(log_gasto_per ~ Area + TIPOVIVIENDA +log_ingreso_per,
          data = encuesta_obs)
```

Modelo para imputación del estado de ocupación

```
mod.mult <- multinom(
              TECHOVIVIENDA ~ Area + TIPOVIVIENDA + log_ingreso_per,
              data = encuesta_obs, trace = FALSE)
```

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$log_gasto_per_imp <- imp
encuesta_no_obs$TECHOVIVIENDA_imp <- imp.mult
encuesta <- bind_rows(encuesta_obs,encuesta_no_obs)
```

Imputación por regresión

Resultados de la imputación

TECHOVIVIENDA	value_missin	value_imp
1. Teja de barro	0.0696	0.0696
2. Teja de cemento	0.0063	0.0063
3. Lámina de asbesto acanalado	0.0319	0.0319
4. Lámina de zinc acanalada	0.2337	0.3003
5. Lámina de aluzinc acanalada	0.3165	0.5500
6. Losa de concreto sin canaletas	0.0214	0.0214
7. Losa de concreto con canaletas	0.0181	0.0181
8. Shingle	0.0013	0.0013
9. Paja, palma y similares	0.0005	0.0005
10. Otro, especifique	0.0008	0.0008
NA	0.3000	0.0000

Imputación por regresión (Resultados antes de la imputación)

TECHOVIVIENDA	1. Urbana	2. Rural	Sum	
1. Teja de barro	0.0503	0.0193	0.0696	0
2. Teja de cemento	0.0047	0.0016	0.0063	0
3. Lámina de asbesto acanalado	0.0310	0.0009	0.0319	0
4. Lámina de zinc acanalada	0.1923	0.0414	0.2337	0
5. Lámina de aluzinc acanalada	0.2788	0.0377	0.3165	0
6. Losa de concreto sin canaletas	0.0212	0.0002	0.0214	0
7. Losa de concreto con canaletas	0.0176	0.0005	0.0181	0
8. Shingle	0.0010	0.0002	0.0013	0
9. Paja, palma y similares	0.0000	0.0005	0.0005	0
10. Otro, especifique	0.0008	0.0000	0.0008	0
Sum	0.8532	0.1468	1.0000	0
NA	0.2555	0.0445	0.3000	0

Imputación por regresión (Resultados después de la imputación)

TECHOVIVIENDA	1. Urbana	2. Rural	Sum	
1. Teja de barro	0.0503	0.0193	0.0696	0
2. Teja de cemento	0.0047	0.0016	0.0063	0
3. Lámina de asbesto acanalado	0.0310	0.0009	0.0319	0
4. Lámina de zinc acanalada	0.2310	0.0693	0.3003	0
5. Lámina de aluzinc acanalada	0.4957	0.0543	0.5500	0
6. Losa de concreto sin canaletas	0.0212	0.0002	0.0214	0
7. Losa de concreto con canaletas	0.0176	0.0005	0.0181	0
8. Shingle	0.0010	0.0002	0.0013	0
9. Paja, palma y similares	0.0000	0.0005	0.0005	0
10. Otro, especifique	0.0008	0.0000	0.0008	0
Sum	0.8532	0.1468	1.0000	0
NA	0.0000	0.0000	0.0000	0

Imputación por regresión (Resultados antes de la imputación)

TIPOVIVIENDA	1. Teja de Abarro	2. Teja de cemento	3. Lámina de asbesto acanalado	4. Lámina de zinc acanalada	5. Lámina de aluzinc acanalada	6. Losa de concreto sin canaletas	7. Losa de concreto con canaletas	8. Shingle	9. Paja, palma y similares	10. Otro, especifique	Sum
1. Casa	0.0667	0.0058	0.0302	0.2094	0.2860	0.0127	0.0110	0.0013	5e-04	7e-04	0.88980.2657
2. Apartamento	0.0008	0.0002	0.0013	0.0103	0.0181	0.0072	0.0059	0.0000	0e+00	0e+00	0.06320.0194
3. Cuarto en mesón o cuartería	0.0021	0.0002	0.0005	0.0136	0.0122	0.0014	0.0010	0.0000	0e+00	1e-04	0.04560.0145
4. Local no construido para vivienda	0.0001	0.0000	0.0000	0.0005	0.0002	0.0001	0.0001	0.0000	0e+00	0e+00	0.00140.0003
5. Otro, especifique	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0e+00	0e+00	0.00000.0000
Sum	0.0696	0.0063	0.0319	0.2337	0.3165	0.0214	0.0181	0.0013	5e-04	8e-04	1.00000.3000
NA	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0e+00	0e+00	0.00000.0000

Imputación por regresión (Resultados después de la imputación)

TIPOVIVIENDA	1. Teja de barro	2. Teja de cemento	3. Lámina de asbesto acanalado	4. Lámina de zinc acanalada	5. Lámina de aluzinc acanalada	6. Losa de concreto sin canaletas	7. Losa de concreto con canaletas	8. Shingle	9. Paja, palma y similares	10. Otro, especifique	Sum	
1. Casa	0.0667	0.0058	0.0302	0.2663	0.4947	0.0127	0.0110	0.0013	5e-04	7e-04	0.8898	0
2. Apartamento	0.0008	0.0002	0.0013	0.0105	0.0373	0.0072	0.0059	0.0000	0e+00	0e+00	0.0632	0
3. Cuarto en mesón o cuartería	0.0021	0.0002	0.0005	0.0226	0.0177	0.0014	0.0010	0.0000	0e+00	1e-04	0.0456	0
4. Local no construido para vivienda	0.0001	0.0000	0.0000	0.0008	0.0002	0.0001	0.0001	0.0000	0e+00	0e+00	0.0014	0
5. Otro, especifique	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0e+00	0e+00	0.0000	0
Sum	0.0696	0.0063	0.0319	0.3003	0.5500	0.0214	0.0181	0.0013	5e-04	8e-04	1.0000	0
NA	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0e+00	0e+00	0.0000	0

Imputación por regresión

```
encuesta %>% summarise(
  log_gasto = mean(log_gasto_per),
  log_gasto_sd = sd(log_gasto_per),
  log_gasto_imp = mean(log_gasto_per_imp),
  log_gasto_imp_sd = sd(log_gasto_per_imp))  %>%
  mutate(BR = 100*(log_gasto - log_gasto_imp)/log_gasto )
```

log_gasto	log_gasto_sd	log_gasto_imp	log_gasto_imp_sd	BR
8.699	0.6876	8.727	0.6701	-0.3202

Imputación por regresión

```
encuesta %>%group_by(Area) %>%  summarise(
  log_gasto = mean(log_gasto_per),
  log_gasto_sd = sd(log_gasto_per),
  log_gasto_imp = mean(log_gasto_per_imp),
  log_gasto_imp_sd = sd(log_gasto_per_imp))  %>%
  mutate(BR = 100*(log_gasto - log_gasto_imp)/log_gasto )
```

Area	log_gasto	log_gasto_sd	log_gasto_imp	log_gasto_imp_sd	BR
1. Urbana	8.762	0.6632	8.793	0.6443	-0.3467
2. Rural	8.333	0.7127	8.346	0.6905	-0.1580

Imputación por regresión

TIPOVIVIENDA	log_gasto	log_gasto_sd	log_gasto_imp	log_gasto_imp_sd	BR
1. Casa	8.684	0.6904	8.717	0.6733	-0.3721
2. Apartamento	8.966	0.6777	8.965	0.6595	0.0091
3. Cuarto en mesón o cuartería	8.623	0.5615	8.605	0.5436	0.2018
4. Local no construido para vivienda	8.481	0.4675	8.438	0.4786	0.5070

Imputación por regresión

```
## Ordenando la base para gráfica
dat_plot7 <- tidyr::gather(
  encuesta %>% dplyr::select(Area,TIPOVIVIENDA,log_gasto_per,
                             log_gasto_per_imp),
  key = "Caso", value = "log_gasto_per2", -Area,-TIPOVIVIENDA)

p1 <- ggplot(dat_plot7, aes(x = log_gasto_per2, fill = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
   theme(legend.position = "bottom") +
  geom_vline(
    xintercept = mean(encuesta$log_gasto_per),
              col = "red") +
  geom_vline(
    xintercept = mean(encuesta$log_gasto_per_imp),
              col = "blue")
p1
```

Imputación por regresión

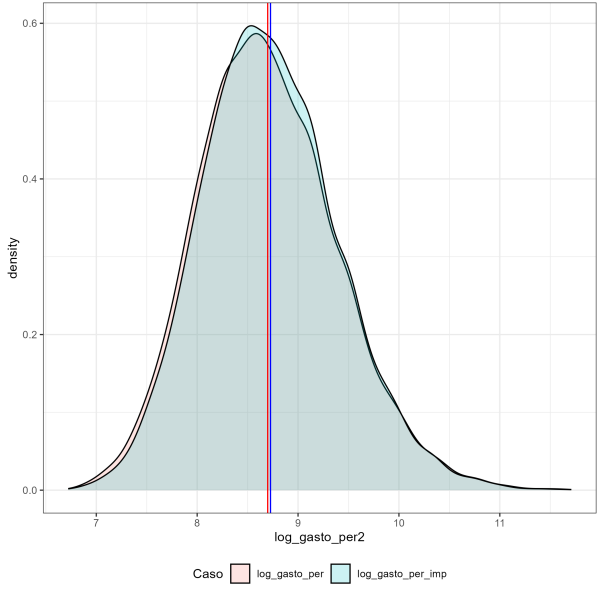


Figura 19: Imputación por regresión

Imputación por regresión

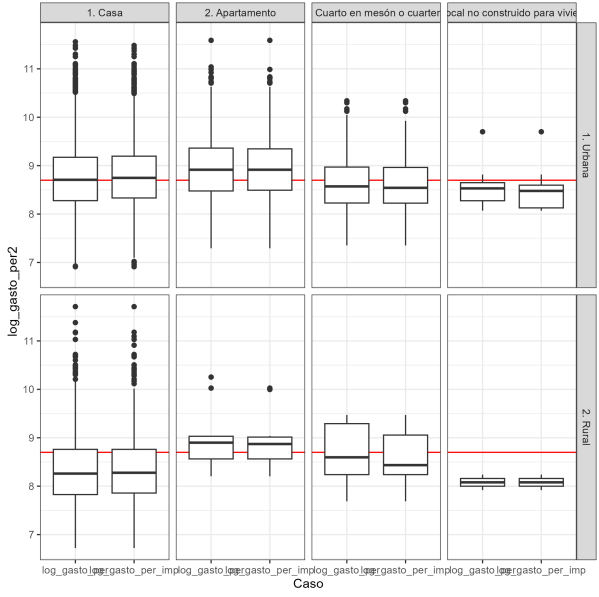


Figura 20: Imputación por regresión por tipo de vivienda y área

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$log_gasto_per_imp <- encuesta$log_gasto_per_missin
encuesta$TECHOVIVIENDA_imp <- encuesta$TECHOVIVIENDA_missin
```

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

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

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

```
encuesta_no_obs <- filter(encuesta,
                          is.na(log_gasto_per_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 log_gasto_per en la fila actual
  Expen_ii <- encuesta_no_obs$log_gasto_per[[ii]]

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

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

  # Asignar el valor de TECHOVIVIENDA_missin correspondiente al índice encontrado
  encuesta_no_obs$TECHOVIVIENDA_imp[[ii]] <-
    encuesta_obs$TECHOVIVIENDA_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 de la imputación

TECHOVIVIENDA	value_missin	value_imp
1. Teja de barro	0.0696	0.1030
2. Teja de cemento	0.0063	0.0087
3. Lámina de asbesto acanalado	0.0319	0.0447
4. Lámina de zinc acanalada	0.2337	0.3340
5. Lámina de aluzinc acanalada	0.3165	0.4501
6. Losa de concreto sin canaletas	0.0214	0.0311
7. Losa de concreto con canaletas	0.0181	0.0249
8. Shingle	0.0013	0.0017
9. Paja, palma y similares	0.0005	0.0006
10. Otro, especifique	0.0008	0.0011
NA	0.3000	0.0000

Imputación por el vecino más cercano (Antes)

TECHOVIVIENDA	1. Urbana	2. Rural	Sum	
1. Teja de barro	0.0503	0.0193	0.0696	0
2. Teja de cemento	0.0047	0.0016	0.0063	0
3. Lámina de asbesto acanalado	0.0310	0.0009	0.0319	0
4. Lámina de zinc acanalada	0.1923	0.0414	0.2337	0
5. Lámina de aluzinc acanalada	0.2788	0.0377	0.3165	0
6. Losa de concreto sin canaletas	0.0212	0.0002	0.0214	0
7. Losa de concreto con canaletas	0.0176	0.0005	0.0181	0
8. Shingle	0.0010	0.0002	0.0013	0
9. Paja, palma y similares	0.0000	0.0005	0.0005	0
10. Otro, especifique	0.0008	0.0000	0.0008	0
Sum	0.8532	0.1468	1.0000	0
NA	0.2555	0.0445	0.3000	0

Imputación por el vecino más cercano (Después)

TECHOVIVIENDA	1. Urbana	2. Rural	Sum	
1. Teja de barro	0.0784	0.0246	0.1030	0
2. Teja de cemento	0.0069	0.0018	0.0087	0
3. Lámina de asbesto acanalado	0.0423	0.0024	0.0447	0
4. Lámina de zinc acanalada	0.2772	0.0568	0.3340	0
5. Lámina de aluzinc acanalada	0.3930	0.0572	0.4501	0
6. Losa de concreto sin canaletas	0.0292	0.0019	0.0311	0
7. Losa de concreto con canaletas	0.0236	0.0014	0.0249	0
8. Shingle	0.0015	0.0002	0.0017	0
9. Paja, palma y similares	0.0001	0.0005	0.0006	0
10. Otro, especifique	0.0011	0.0000	0.0011	0
Sum	0.8532	0.1468	1.0000	0
NA	0.0000	0.0000	0.0000	0

Imputación por el vecino más cercano (Antes)

TECHOVIVIENDA	1. Casa	2. Apartamento	3. Cuarto en mesón o cuartería	4. Local no construido para vivienda	5. Otro, especifique	Sum	
1. Teja de barro	0.0667	0.0008	0.0021	0.0001	0	0.0696	0
2. Teja de cemento	0.0058	0.0002	0.0002	0.0000	0	0.0063	0
3. Lámina de asbesto acanalado	0.0302	0.0013	0.0005	0.0000	0	0.0319	0
4. Lámina de zinc acanalada	0.2094	0.0103	0.0136	0.0005	0	0.2337	0
5. Lámina de aluzinc acanalada	0.2860	0.0181	0.0122	0.0002	0	0.3165	0
6. Losa de concreto sin canaletas	0.0127	0.0072	0.0014	0.0001	0	0.0214	0
7. Losa de concreto con canaletas	0.0110	0.0059	0.0010	0.0001	0	0.0181	0
8. Shingle	0.0013	0.0000	0.0000	0.0000	0	0.0013	0
9. Paja, palma y similares	0.0005	0.0000	0.0000	0.0000	0	0.0005	0
10. Otro, especifique	0.0007	0.0000	0.0001	0.0000	0	0.0008	0
Sum	0.8898	0.0632	0.0456	0.0014	0	1.0000	0
NA	0.2657	0.0194	0.0145	0.0003	0	0.3000	0

Imputación por el vecino más cercano (Después)

TECHOVIVIENDA	1. Casa	2. Apartamento	3. Cuarto en mesón o cuartería	4. Local no construido para vivienda	5. Otro, especifique	Sum	
1. Teja de barro	0.0962	0.0029	0.0039	0.0001	0	0.1030	0
2. Teja de cemento	0.0080	0.0005	0.0002	0.0000	0	0.0087	0
3. Lámina de asbesto acanalado	0.0415	0.0024	0.0007	0.0001	0	0.0447	0
4. Lámina de zinc acanalada	0.2977	0.0177	0.0180	0.0006	0	0.3340	0
5. Lámina de aluzinc acanalada	0.4053	0.0256	0.0189	0.0003	0	0.4501	0
6. Losa de concreto sin canaletas	0.0208	0.0078	0.0024	0.0001	0	0.0311	0
7. Losa de concreto con canaletas	0.0172	0.0062	0.0015	0.0001	0	0.0249	0
8. Shingle	0.0016	0.0001	0.0000	0.0000	0	0.0017	0
9. Paja, palma y similares	0.0006	0.0000	0.0000	0.0000	0	0.0006	0
10. Otro, especifique	0.0009	0.0001	0.0001	0.0000	0	0.0011	0
Sum	0.8898	0.0632	0.0456	0.0014	0	1.0000	0
NA	0.0000	0.0000	0.0000	0.0000	0	0.0000	0

Imputación por el vecino más cercano

```
encuesta %>% summarise(
  log_gasto = mean(log_gasto_per),
  log_gasto_sd = sd(log_gasto_per),
  log_gasto_imp = mean(log_gasto_per_imp),
  log_gasto_imp_sd = sd(log_gasto_per_imp))%>%
  mutate(BR = 100*(log_gasto - log_gasto_imp)/log_gasto)
```

log_gasto	log_gasto_sd	log_gasto_imp	log_gasto_imp_sd	BR
8.699	0.6876	8.699	0.6877	0

Imputación por el vecino más cercano

```
encuesta %>%group_by(Area) %>%   summarise(
  log_gasto = mean(log_gasto_per),
  log_gasto_sd = sd(log_gasto_per),
  log_gasto_imp = mean(log_gasto_per_imp),
  log_gasto_imp_sd = sd(log_gasto_per_imp))%>%
  mutate(BR = 100*(log_gasto - log_gasto_imp)/log_gasto)
```

Area	log_gasto	log_gasto_sd	log_gasto_imp	log_gasto_imp_sd	BR
1. Urbana	8.762	0.6632	8.762	0.6632	0e+00
2. Rural	8.333	0.7127	8.333	0.7129	-3e-04

Imputación por el vecino más cercano

TIPOVIVIENDA	log_gasto	log_gasto_sd	log_gasto_imp	log_gasto_imp_sd	BR
1. Casa	8.684	0.6904	8.684	0.6905	0e+00
2. Apartamento	8.966	0.6777	8.966	0.6777	0e+00
3. Cuarto en mesón o cuartería	8.623	0.5615	8.623	0.5615	-2e-04
4. Local no construido para vivienda	8.481	0.4675	8.481	0.4675	3e-04

Imputación por el vecino más cercano

```
## Ordenando la base para gráfica
dat_plot8 <- tidyr::gather(
  encuesta %>% dplyr::select(Area,TIPOVIVIENDA,log_gasto_per,
                             log_gasto_per_imp),
  key = "Caso", value = "log_gasto_per2", -Area,-TIPOVIVIENDA)

p1 <- ggplot(dat_plot8, aes(x = log_gasto_per2, fill = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
   theme(legend.position = "bottom") +
  geom_vline(
    xintercept = mean(encuesta$log_gasto_per),
              col = "red") +
  geom_vline(
    xintercept = mean(encuesta$log_gasto_per_imp),
              col = "blue")
p1
```

Imputación por el vecino más cercano

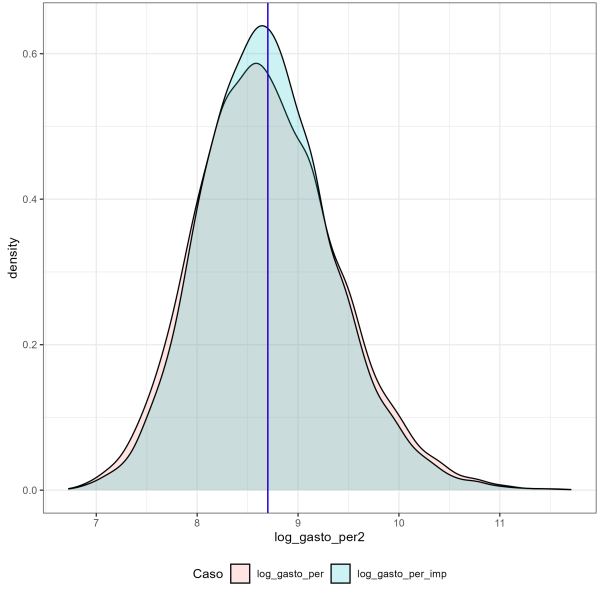


Figura 21: Imputación por el vecino más cercano

Imputación por el vecino más cercano

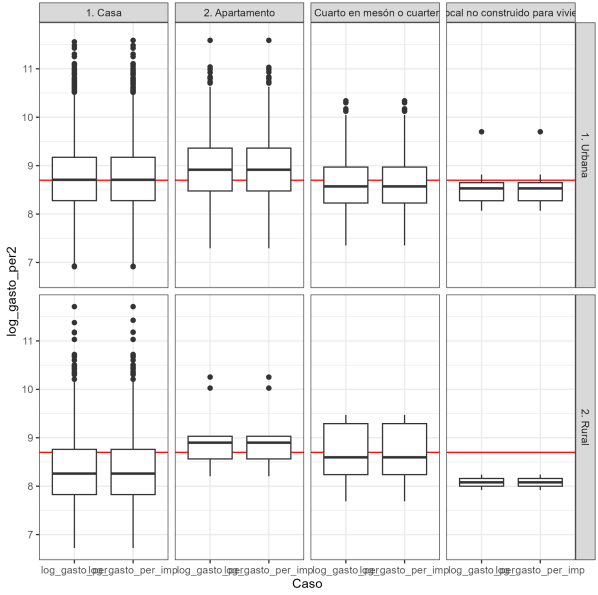


Figura 22: Imputación por el vecino más cercano por TIPOVIVIENDAo y zona

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

```
# Imputación de valores faltantes en las columnas
# 'log_gasto_per_imp' y 'TECHOVIVIENDA_imp'
encuesta$log_gasto_per_imp <- encuesta$log_gasto_per_missin
encuesta$TECHOVIVIENDA_imp <- encuesta$TECHOVIVIENDA_missin

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

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

# Ajuste de un modelo de regresión lineal utilizando las observaciones con
mod <- lm(log_gasto_per ~ Area + TIENEVEHICULOS +log_ingreso_per  ,
          data = encuesta_obs)
```

Imputación por el vecino más cercano con regresión

```r
# Predicciones para las observaciones con 'log_gasto_per_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$log_gasto_per_imp[[ii]] <-
    encuesta_obs$log_gasto_per_missin[[don_ii]]
  encuesta_no_obs$TECHOVIVIENDA_imp[[ii]] <-
    encuesta_obs$TECHOVIVIENDA_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 de la imputación

TECHOVIVIENDA	value_missin	value_imp
1. Teja de barro	0.0696	0.1046
2. Teja de cemento	0.0063	0.0089
3. Lámina de asbesto acanalado	0.0319	0.0465
4. Lámina de zinc acanalada	0.2337	0.3259
5. Lámina de aluzinc acanalada	0.3165	0.4538
6. Losa de concreto sin canaletas	0.0214	0.0308
7. Losa de concreto con canaletas	0.0181	0.0256
8. Shingle	0.0013	0.0021
9. Paja, palma y similares	0.0005	0.0005
10. Otro, especifique	0.0008	0.0014
NA	0.3000	0.0000

Imputación por el vecino más cercano con regresión (Antes)

TECHOVIVIENDA	1. Urbana	2. Rural	Sum	
1. Teja de barro	0.0503	0.0193	0.0696	0
2. Teja de cemento	0.0047	0.0016	0.0063	0
3. Lámina de asbesto acanalado	0.0310	0.0009	0.0319	0
4. Lámina de zinc acanalada	0.1923	0.0414	0.2337	0
5. Lámina de aluzinc acanalada	0.2788	0.0377	0.3165	0
6. Losa de concreto sin canaletas	0.0212	0.0002	0.0214	0
7. Losa de concreto con canaletas	0.0176	0.0005	0.0181	0
8. Shingle	0.0010	0.0002	0.0013	0
9. Paja, palma y similares	0.0000	0.0005	0.0005	0
10. Otro, especifique	0.0008	0.0000	0.0008	0
Sum	0.8532	0.1468	1.0000	0
NA	0.2555	0.0445	0.3000	0

Imputación por el vecino más cercano con regresión (Después)

TECHOVIVIENDA	1. Urbana	2. Rural	Sum	
1. Teja de barro	0.0798	0.0248	0.1046	0
2. Teja de cemento	0.0069	0.0021	0.0089	0
3. Lámina de asbesto acanalado	0.0440	0.0025	0.0465	0
4. Lámina de zinc acanalada	0.2693	0.0566	0.3259	0
5. Lámina de aluzinc acanalada	0.3958	0.0580	0.4538	0
6. Losa de concreto sin canaletas	0.0297	0.0010	0.0308	0
7. Losa de concreto con canaletas	0.0245	0.0011	0.0256	0
8. Shingle	0.0018	0.0002	0.0021	0
9. Paja, palma y similares	0.0000	0.0005	0.0005	0
10. Otro, especifique	0.0014	0.0000	0.0014	0
Sum	0.8532	0.1468	1.0000	0
NA	0.0000	0.0000	0.0000	0

Imputación por el vecino más cercano con regresión (Antes)

TIPOVIVIENDA	1. Casa	2. Apartamento	3. Cuarto en mesón o cuartería	4. Local no construido para vivienda	5. Otro, especifique	Sum	
1. Teja de barro	0.0667	0.0008	0.0021	0.0001	0	0.0696	0
2. Teja de cemento	0.0058	0.0002	0.0002	0.0000	0	0.0063	0
3. Lámina de asbesto acanalado	0.0302	0.0013	0.0005	0.0000	0	0.0319	0
4. Lámina de zinc acanalada	0.2094	0.0103	0.0136	0.0005	0	0.2337	0
5. Lámina de aluzinc acanalada	0.2860	0.0181	0.0122	0.0002	0	0.3165	0
6. Losa de concreto sin canaletas	0.0127	0.0072	0.0014	0.0001	0	0.0214	0
7. Losa de concreto con canaletas	0.0110	0.0059	0.0010	0.0001	0	0.0181	0
8. Shingle	0.0013	0.0000	0.0000	0.0000	0	0.0013	0
9. Paja, palma y similares	0.0005	0.0000	0.0000	0.0000	0	0.0005	0
10. Otro, especifique	0.0007	0.0000	0.0001	0.0000	0	0.0008	0
Sum	0.8898	0.0632	0.0456	0.0014	0	1.0000	0
NA	0.2657	0.0194	0.0145	0.0003	0	0.3000	0

Imputación por el vecino más cercano con regresión (Después)

TIPOVIVIENDA	1. Casa	2. Apartamento	3. Cuarto en mesón o cuartería	4. Local no construido para vivienda	5. Otro, especifique	Sum
1. Teja de barro	0.0975	0.0027	0.0041	0.0002	0	0.10460
2. Teja de cemento	0.0080	0.0006	0.0003	0.0000	0	0.00890
3. Lámina de asbesto acanalado	0.0432	0.0025	0.0008	0.0000	0	0.04650
4. Lámina de zinc acanalada	0.2912	0.0157	0.0184	0.0006	0	0.32590
5. Lámina de aluzinc acanalada	0.4069	0.0280	0.0185	0.0003	0	0.45380
6. Losa de concreto sin canaletas	0.0212	0.0077	0.0018	0.0001	0	0.03080
7. Losa de	0.0180	0.0061	0.0015	0.0001	0	0.02560

Imputación por el vecino más cercano con regresión

```
encuesta %>% summarise(
  log_gasto = mean(log_gasto_per),
  log_gasto_sd = sd(log_gasto_per),
  log_gasto_imp = mean(log_gasto_per_imp),
  log_gasto_imp_sd = sd(log_gasto_per_imp))%>%
  mutate(BR = 100*(log_gasto - log_gasto_imp)/log_gasto)
```

log_gasto	log_gasto_sd	log_gasto_imp	log_gasto_imp_sd	BR
8.699	0.6876	8.699	0.6912	-0.0038

Imputación por el vecino más cercano con regresión

```
encuesta %>%group_by(Area) %>% summarise(
  log_gasto = mean(log_gasto_per),
  log_gasto_sd = sd(log_gasto_per),
  log_gasto_imp = mean(log_gasto_per_imp),
  log_gasto_imp_sd = sd(log_gasto_per_imp))%>%
  mutate(BR = 100*(log_gasto - log_gasto_imp)/log_gasto)
```

Area	log_gasto	log_gasto_sd	log_gasto_imp	log_gasto_imp_sd	BR
1. Urbana	8.762	0.6632	8.761	0.6692	0.0075
2. Rural	8.333	0.7127	8.339	0.7066	-0.0728

Imputación por el vecino más cercano con regresión

TIPOVIVIENDA	log_gasto	log_gasto_sd	log_gasto_imp	log_gasto_imp_sd	BR
1. Casa	8.684	0.6904	8.689	0.6954	-0.0484
2. Apartamento	8.966	0.6777	8.929	0.6702	0.4152
3. Cuarto en mesón o cuartería	8.623	0.5615	8.600	0.5711	0.2663
4. Local no construido para vivienda	8.481	0.4675	8.475	0.4780	0.0700

Imputación por el vecino más cercano con regresión

```
## Ordenando la base para gráfica
dat_plot9 <- tidyr::gather(
  encuesta %>% dplyr::select(Area, TIPOVIVIENDA, log_gasto_per,
                             log_gasto_per_imp),
  key = "Caso",
  value = "log_gasto_per2",
  -Area,
  -TIPOVIVIENDA
)

p1 <- ggplot(dat_plot9, aes(x = log_gasto_per2, fill = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
  theme(legend.position = "bottom") +
  geom_vline(xintercept = mean(encuesta$log_gasto_per),
             col = "red") +
  geom_vline(xintercept = mean(encuesta$log_gasto_per_imp),
             col = "blue")
p1
```

Imputación por el vecino más cercano con regresión

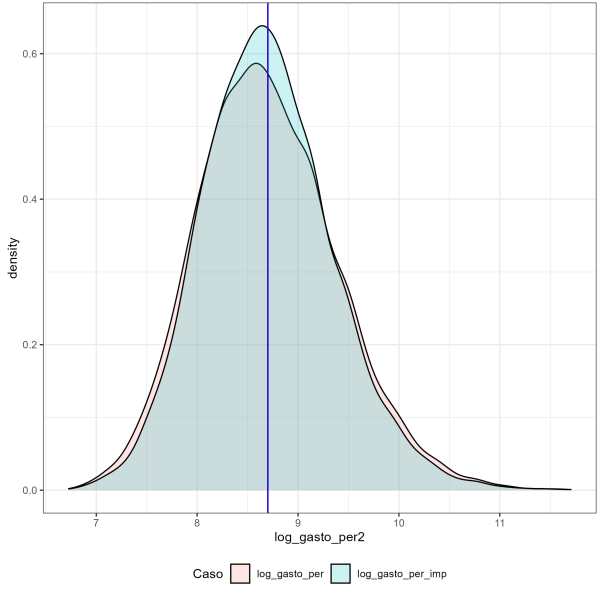


Figura 23: Imputación por el vecino más cercano con regresión

Imputación por el vecino más cercano con regresión

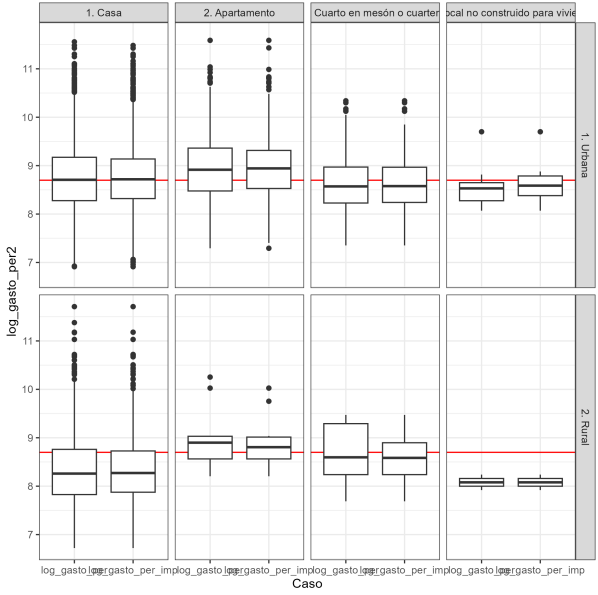


Figura 24: Imputación por el vecino más cercano con regresión

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.

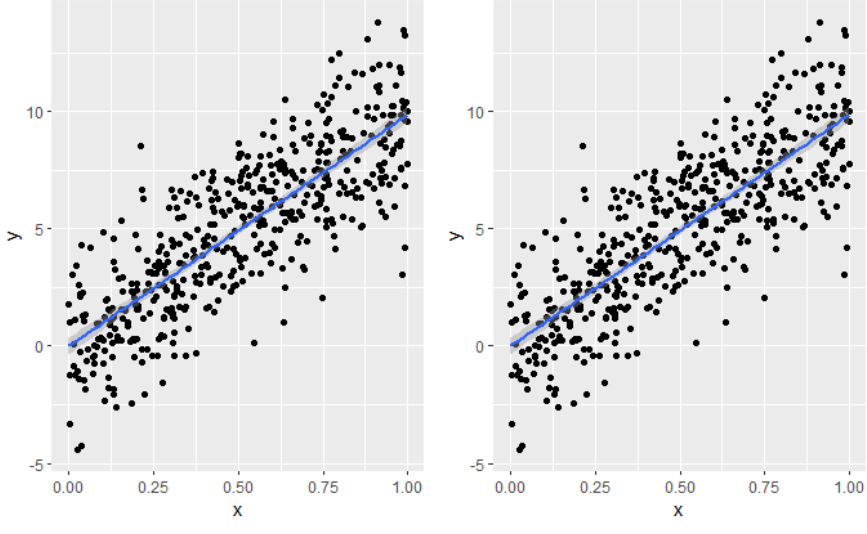


Figura 25: Imputación múltiple

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.

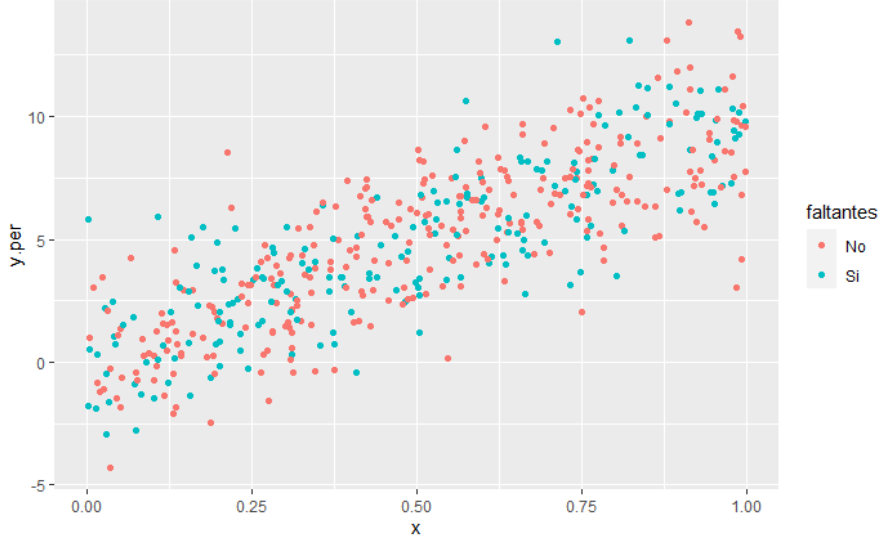


Figura 26: Regresión después de la imputando

Imputación Bootstrap en la encuesta.

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

```
encuesta$log_gasto_per_imp <- encuesta$log_gasto_per_missin
encuesta$TECHOVIVIENDA_imp <- encuesta$TECHOVIVIENDA_missin
encuesta_obs <- filter(encuesta,
                       !is.na(log_gasto_per_missin))
encuesta_no_obs <- filter(encuesta,
                          is.na(log_gasto_per_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("log_gasto_per_vp_", ii)
  vp2 <- paste0("TECHOVIVIENDA_vp_", ii)

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

  mod <- lm(log_gasto_per ~ Area + TIPOVIVIENDA + log_ingreso_per,
            data = encuesta_temp)
  mod.mult <- multinom(TECHOVIVIENDA ~ Area + TIPOVIVIENDA + log_ingreso_per,
                       data = encuesta_temp, )

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

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

```
# weights:  80 (63 variable)
initial  value 14096.425939
```

Imputación Bootstrap en la encuesta.

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

```
dplyr::select(encuesta_no_obs,
       log_gasto_per, matches("log_gasto_per_vp_"))[1:10,1:4]
```

log_gasto_per	log_gasto_per_vp_1	log_gasto_per_vp_2	log_gasto_per_vp_3
8.887	6.927	6.931	6.893
8.304	7.041	7.046	7.007
8.056	7.041	7.046	7.007
9.479	7.123	7.128	7.091
7.096	7.171	7.176	7.140
9.326	7.106	7.111	7.076
7.966	7.282	7.287	7.253
7.650	7.173	7.178	7.145
8.091	7.186	7.191	7.158
7.478	7.214	7.218	7.186

Imputación Bootstrap en la encuesta.

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

```
## Ordenando la base para gráfica
dat_plot10 <- tidyr::gather(
  encuesta %>% dplyr::select(Area,TIPOVIVIENDA,
                             matches("log_gasto_per_vp_")),
  key = "Caso", value = "log_gasto_per2", -Area,-TIPOVIVIENDA)

p1 <- ggplot(dat_plot10, aes(x = log_gasto_per2, col = Caso)) +
  geom_density(alpha = 0.2) + theme_bw() +
   theme(legend.position = "bottom") +
  geom_density(data = encuesta ,aes(x = log_gasto_per),
               col =  "black", size = 1.2)

p1
```

Imputación por el vecino más cercano

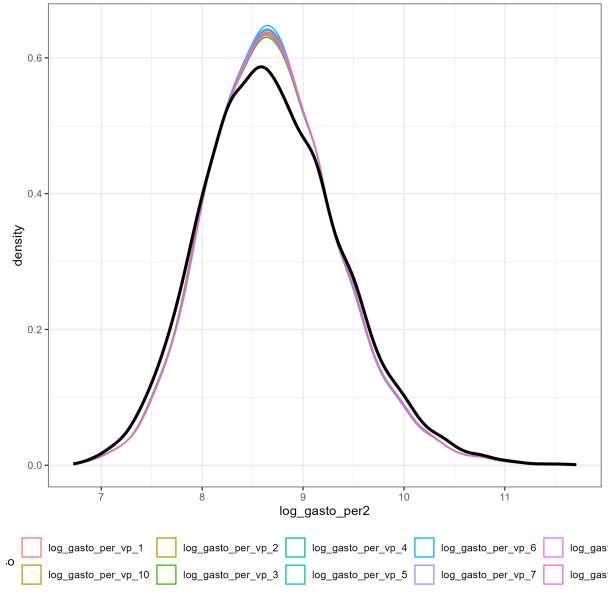


Figura 27: Densidad para los 10 valores plausible

Imputación Bootstrap en la encuesta.

```
## Ordenando la base para gráfica
dat_plot11 <- tidyr::gather(
  encuesta %>%
  dplyr::select(Area,TIPOVIVIENDA, TECHOVIVIENDA,
                matches("TECHOVIVIENDA_vp_")),
  key = "Caso", value = "TECHOVIVIENDA2", -Area,-TIPOVIVIENDA) %>%
  group_by(Caso,TECHOVIVIENDA2) %>% tally() %>%
  group_by(Caso) %>% mutate(prop = n/sum(n))

p1 <- ggplot(dat_plot11,
        aes(x = TECHOVIVIENDA2, 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("TECHOVIVIENDA" = "black"))
p1
```

Imputación por el vecino más cercano

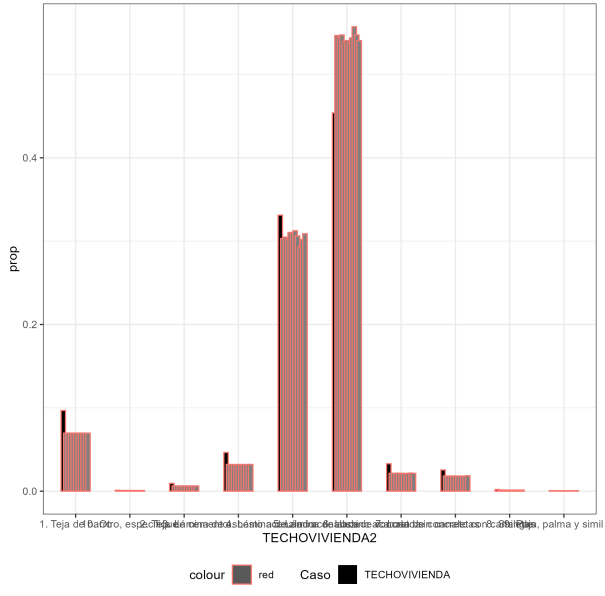


Figura 28: Regresión después de la imputando

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 = estrato,  # Id de los estratos.
    ids = F1_A0_UPM,          # Id para las observaciones.
    weights = Factor,       # Factores de expansión.
    nest = TRUE            # Valida el anidado dentro del estrato
  )
```

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(log_gasto_per_vp_1, vartype = c("var")),
   vp2 = survey_mean(log_gasto_per_vp_2, vartype = c("var")),
   vp3 = survey_mean(log_gasto_per_vp_3, vartype = c("var")),
   vp4 = survey_mean(log_gasto_per_vp_4, vartype = c("var")),
   vp5 = survey_mean(log_gasto_per_vp_5, vartype = c("var")),
   vp6 = survey_mean(log_gasto_per_vp_6, vartype = c("var")),
   vp7 = survey_mean(log_gasto_per_vp_7, vartype = c("var")),
   vp8 = survey_mean(log_gasto_per_vp_8, vartype = c("var")),
   vp9 = survey_mean(log_gasto_per_vp_9, vartype = c("var")),
   vp10 =survey_mean(log_gasto_per_vp_10, vartype = c("var")))
```

Estimación del promedio con valores plausibles (vp)

vp	promedio	var
1	8.569	3e-04
2	8.573	3e-04
3	8.571	3e-04
4	8.570	3e-04
5	8.573	3e-04
6	8.572	3e-04
7	8.566	3e-04
8	8.573	3e-04
9	8.572	3e-04
10	8.569	3e-04

Estimación del promedio con valores plausibles (vp)

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

```
[1] 0.0002979
```

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

```
[1] 5.708e-06
```

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

Media_vp	Media_vp_se
8.571	1.049

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("log_gasto_per_vp")),
               survey_var,  vartype = "var" )
```

vp	promedio	var
1	0.4355	2e-04
2	0.4361	2e-04
3	0.4402	2e-04
4	0.4304	2e-04
5	0.4271	2e-04
6	0.4328	2e-04
7	0.4290	2e-04
8	0.4368	2e-04
9	0.4341	2e-04
10	0.4376	2e-04

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] 0.0001713
```

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

```
[1] 1.696e-05
```

```
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(log_gasto_per),
                     Var = survey_var(log_gasto_per))
```

Media	Media_se	Var	Var_se
8.562	0.0183	0.4828	0.0146

```
resultado
```

Media_vp	Media_vp_se	var_vp	var_vp_se
8.571	1.049	0.434	0.0138

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("TECHOVIVIENDA_vp_",1:10),
       function(vp){
        diseno %>%
         group_by_at(vars(TECHOVIVIENDA = 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:

TECHOVIVIENDA	vp	prop	prop_var
1. Teja de barro	1	0.0848	1e-04
1. Teja de barro	2	0.0848	1e-04
1. Teja de barro	3	0.0848	1e-04
1. Teja de barro	4	0.0848	1e-04
1. Teja de barro	5	0.0848	1e-04
1. Teja de barro	6	0.0848	1e-04
1. Teja de barro	7	0.0848	1e-04
1. Teja de barro	8	0.0848	1e-04
1. Teja de barro	9	0.0848	1e-04
1. Teja de barro	10	0.0848	1e-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(TECHOVIVIENDA) %>%
  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 (Antes)

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

TECHOVIVIENDA	prop	prop_var
1. Teja de barro	0.1217	1e-04
2. Teja de cemento	0.0104	0e+00
3. Lámina de asbesto acanalado	0.0340	0e+00
4. Lámina de zinc acanalada	0.3522	2e-04
5. Lámina de aluzinc acanalada	0.4395	2e-04
6. Losa de concreto sin canaletas	0.0197	0e+00
7. Losa de concreto con canaletas	0.0206	0e+00
8. Shingle	0.0011	0e+00
9. Paja, palma y similares	0.0003	0e+00
10. Otro, especifique	0.0004	0e+00

Comparando resultados con valores plausibles (Después)

```
resultado
```

TECHOVIVIENDA	prop_pv	Ubar	B	prop_pv_var
1. Teja de barro	0.0848	1e-04	0e+00	1e-04
2. Teja de cemento	0.0077	0e+00	0e+00	0e+00
3. Lámina de asbesto acanalado	0.0233	0e+00	0e+00	0e+00
4. Lámina de zinc acanalada	0.3600	2e-04	1e-04	2e-04
5. Lámina de aluzinc acanalada	0.4952	2e-04	1e-04	2e-04
6. Losa de concreto sin canaletas	0.0125	0e+00	0e+00	0e+00
7. Losa de concreto con canaletas	0.0154	0e+00	0e+00	0e+00
8. Shingle	0.0006	0e+00	0e+00	0e+00
9. Paja, palma y similares	0.0002	0e+00	0e+00	0e+00
10. Otro, especifique	0.0003	0e+00	0e+00	0e+00

¡Gracias!

Email: andres.gutierrez@cepal.org