Imputación

Stalyn Guerrero

21/3/2022

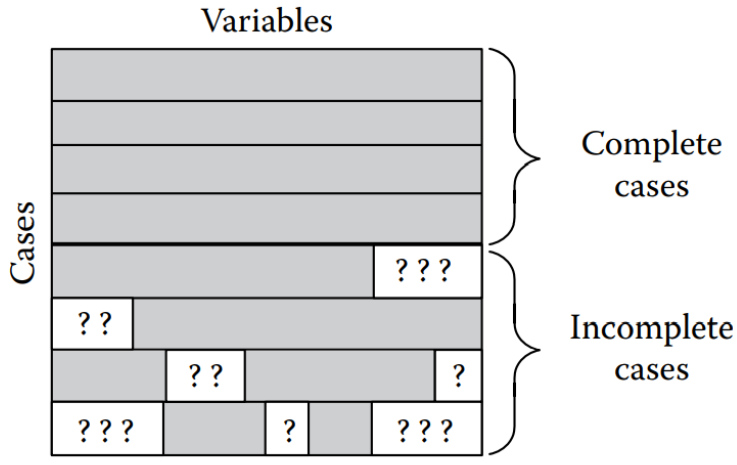
Introducción valores perdidos

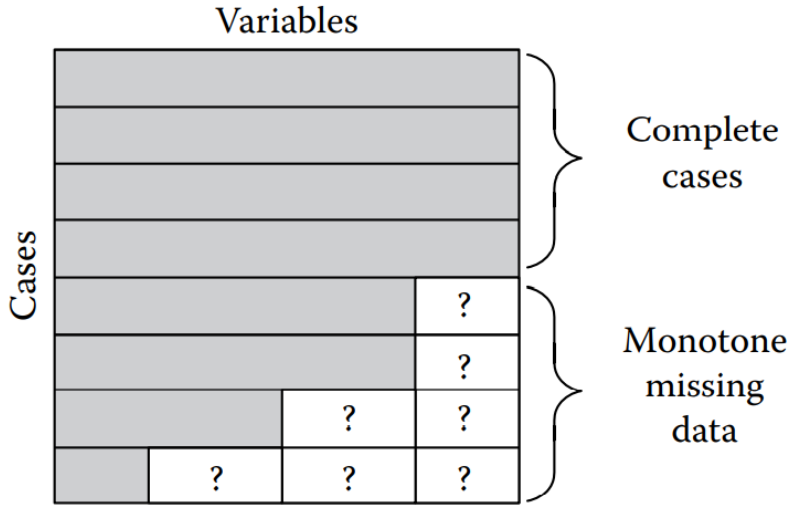
- Sea $\boldsymbol{X}_{n\times p} = x_{ij}$ una matriz completa (sin valores perdidos), de tal forma que X_{ij} es el valor de la variable j, $j = 1, \ldots, p$ en el caso i, $i = 1, \ldots, n$.
- Sea $\boldsymbol{M}_{n\times p} = m_{ij}$, donde $m_{ij} = 1$ si x_{ij} es un dato perdido y $m_{ij} = 0$ 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 $\boldsymbol{M}_{n\times p}$ 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 $\boldsymbol{M}_{n\times p}$ 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





Lectura de la base

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

Unemployed	Inactive	Employed
0.0461	0.3994	0.5545

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

```
## [1] 537.6
```

Creando valores perdidos

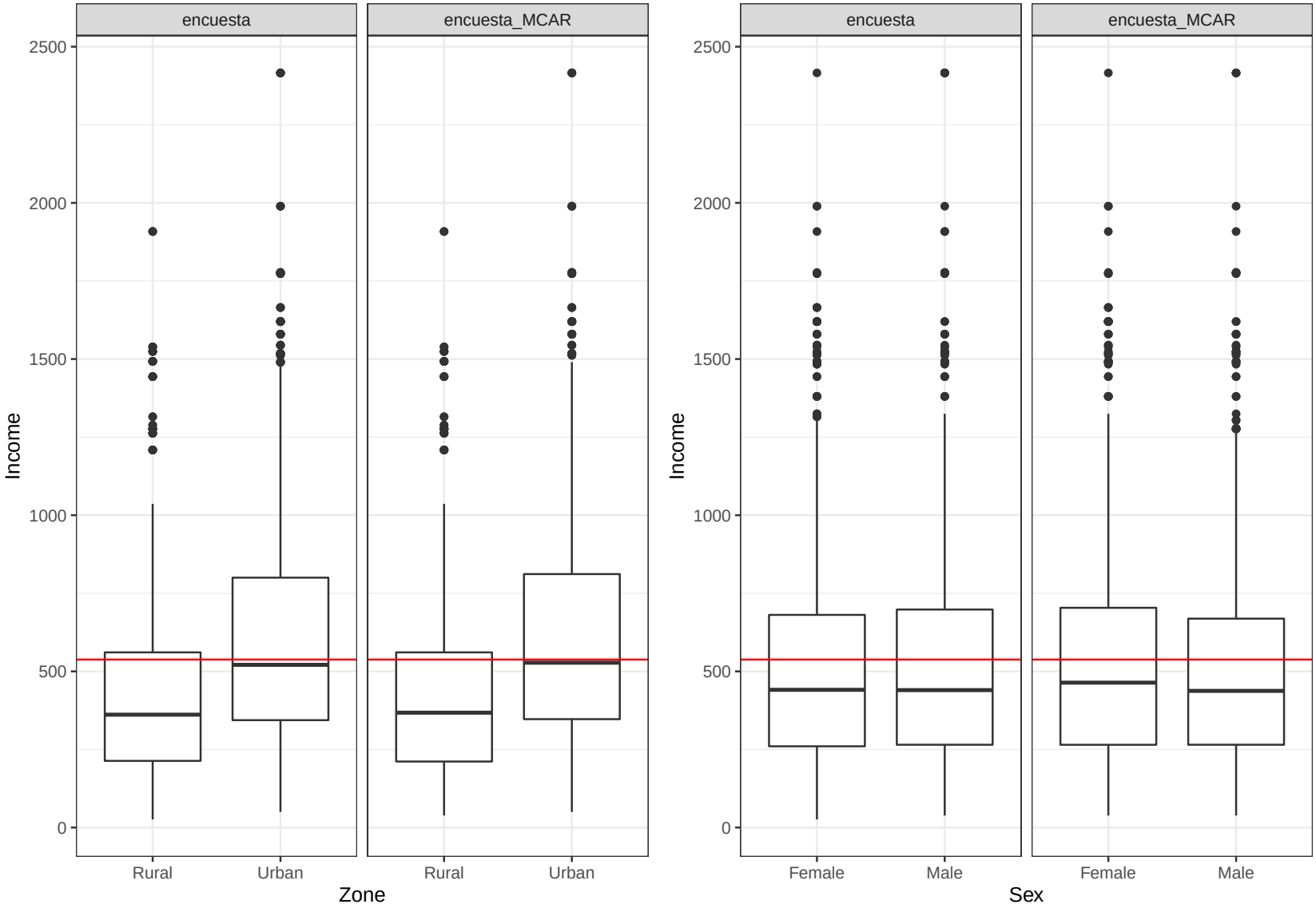
```
set.seed(1234)
encuesta_MCAR <-  sample_frac(encuesta, 0.8 )
dat_plot <- bind_rows(
  list(encuesta_MCAR = encuesta_MCAR,
       encuesta = encuesta), .id = "Caso"  )
```

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

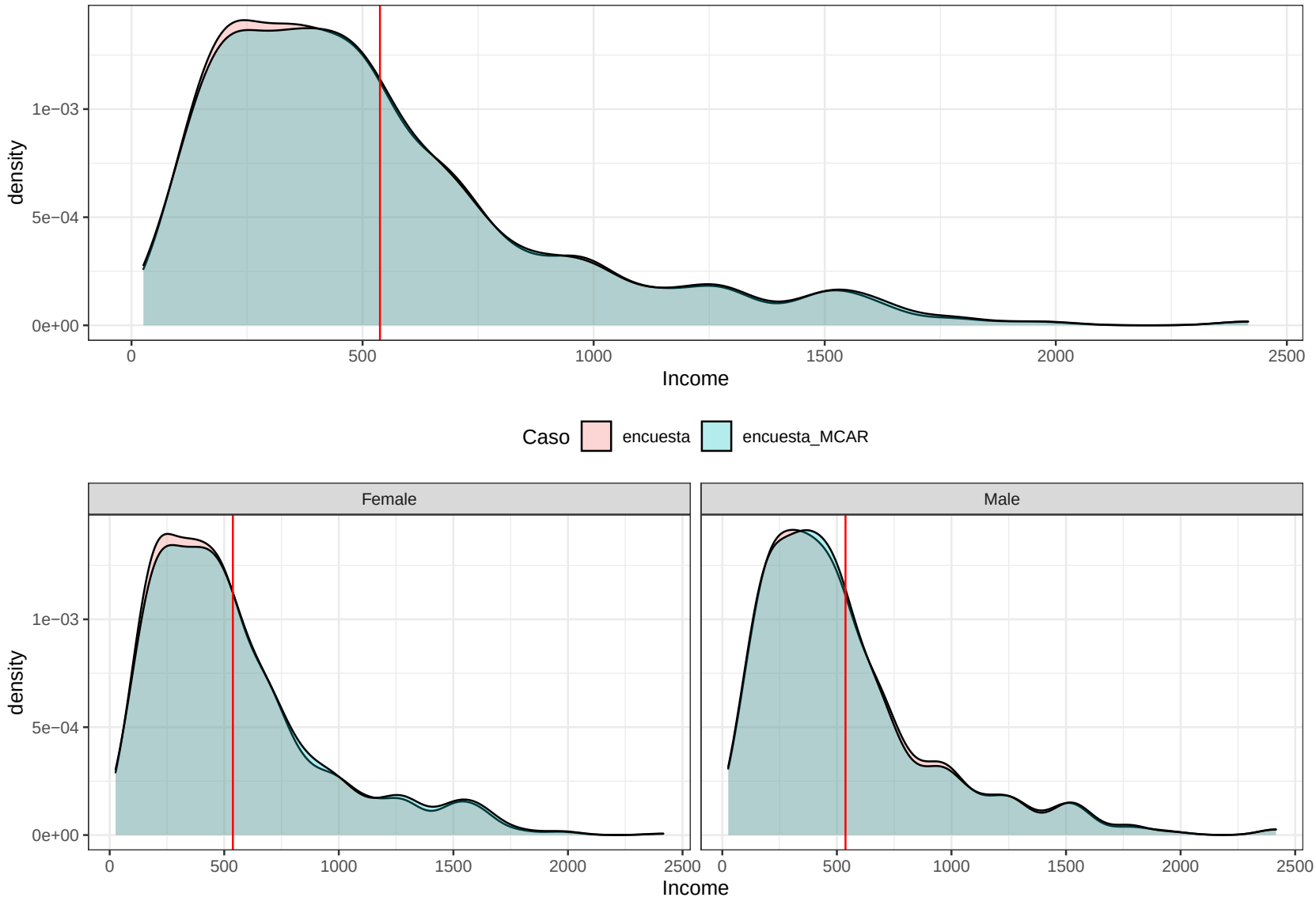


Creando valores perdidos

```
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

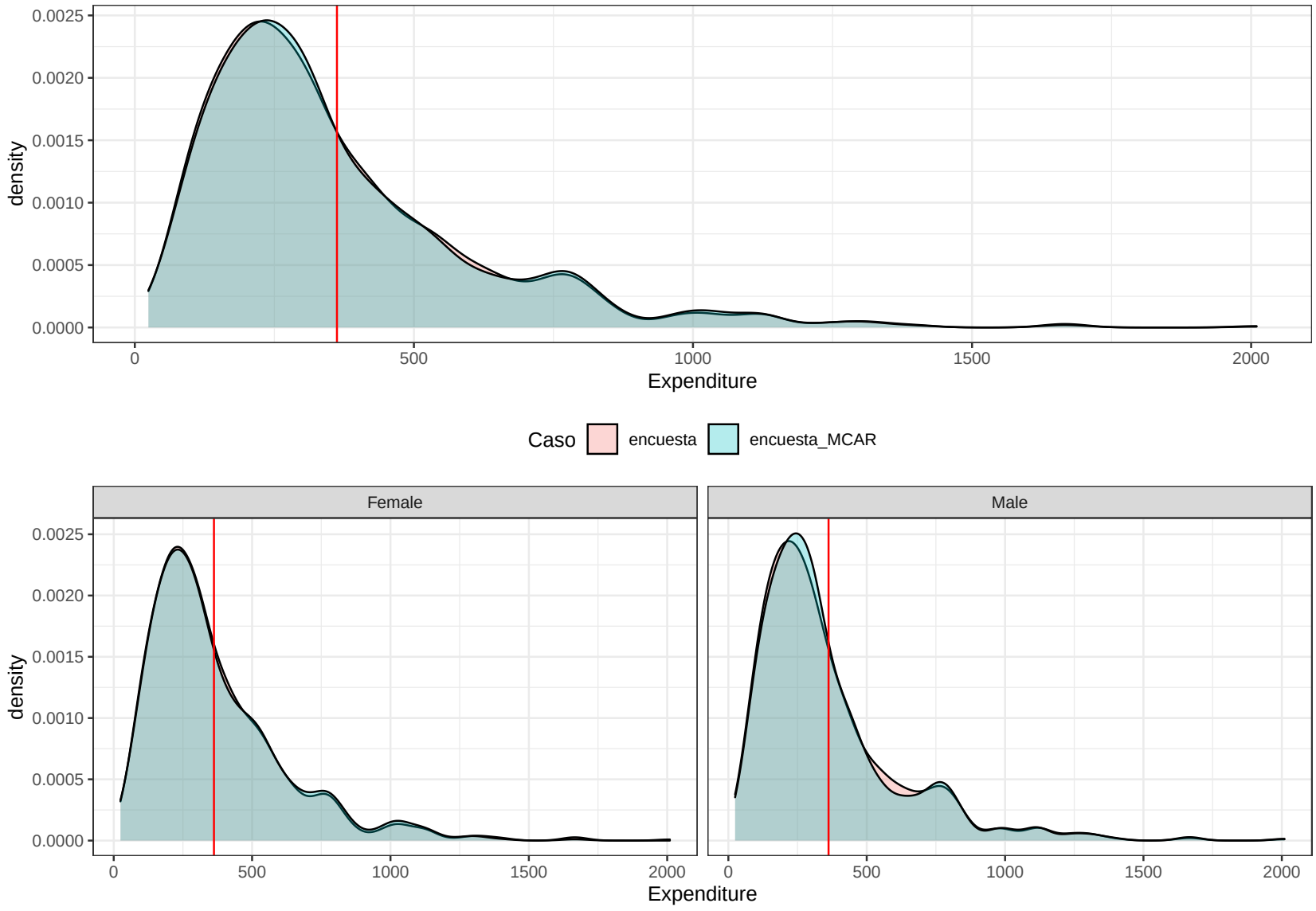


Creando valores perdidos

```
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



Creando valores perdidos

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

RuralFemale	RuralMale	UrbanFemale	UrbanMale
469	411	510	390

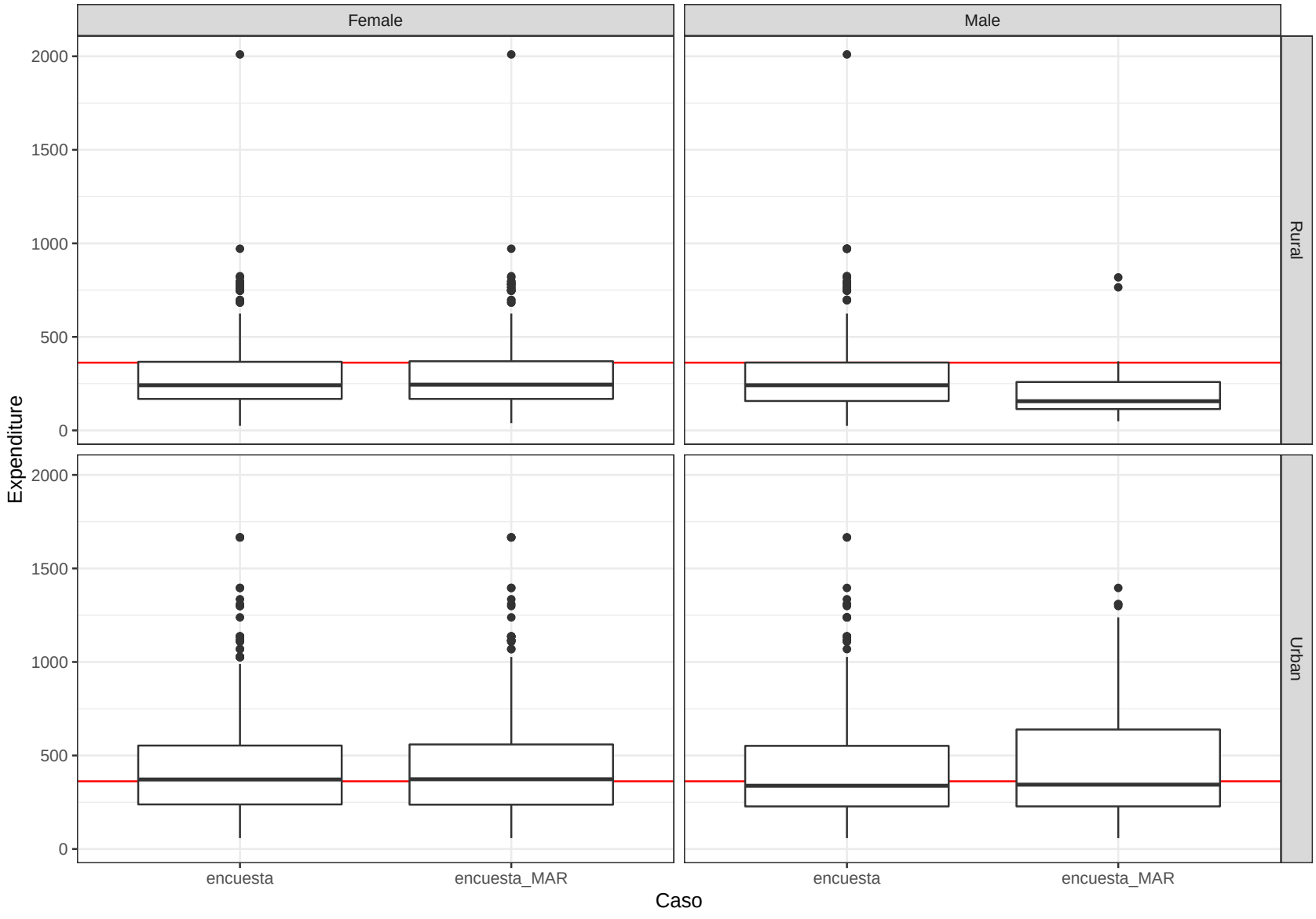
```
sel <- S.STSI(S = temp_estrato,
              Nh = c(469,411,510,390),
              nh = c(20, 380, 20,280))
encuesta_MAR <- encuesta[-sel,]
dat_plot2 <- bind_rows(
  list(encuesta_MAR = encuesta_MAR,
       encuesta = encuesta), .id = "Caso"  )
```

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

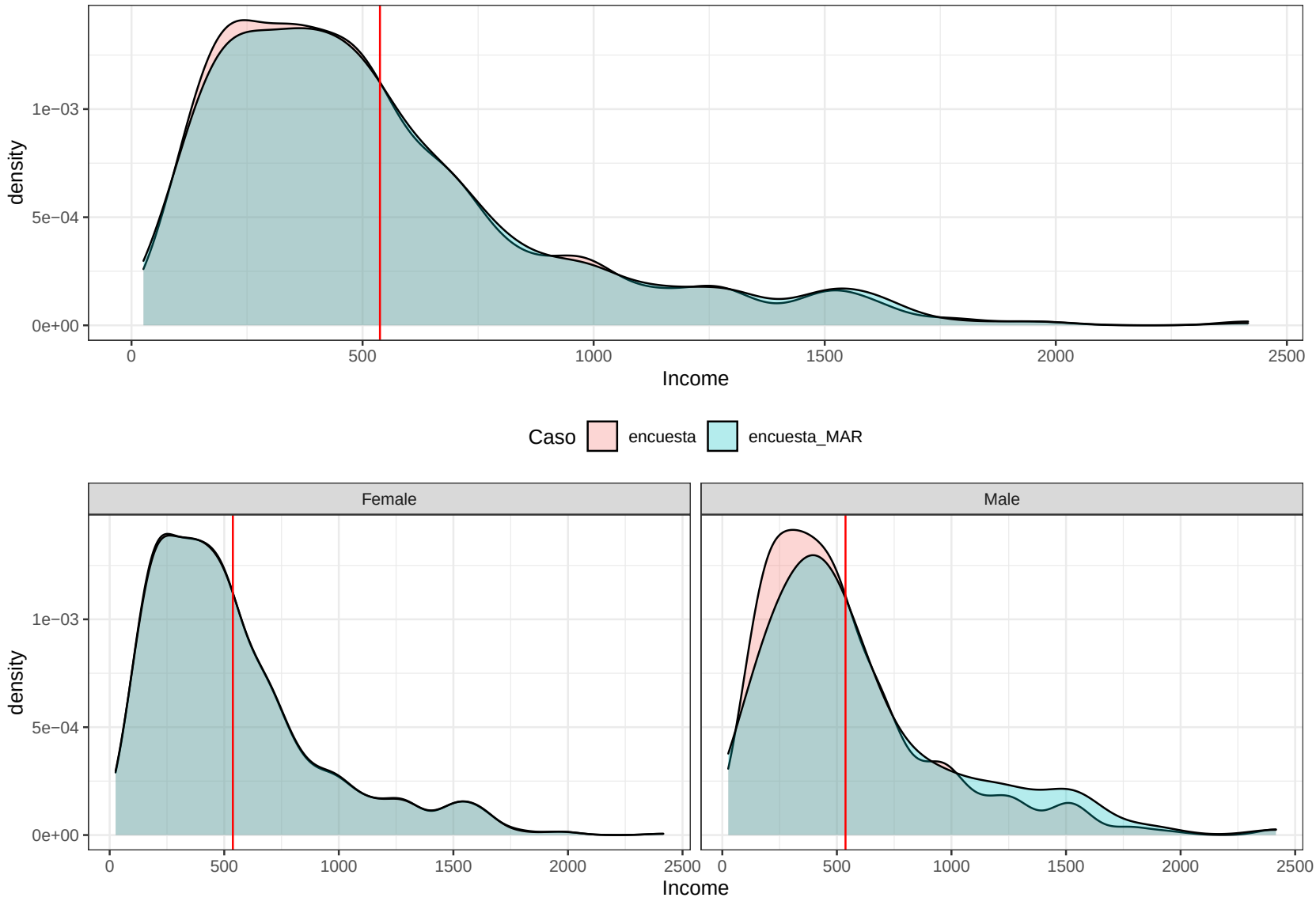


Creando valores perdidos

```
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

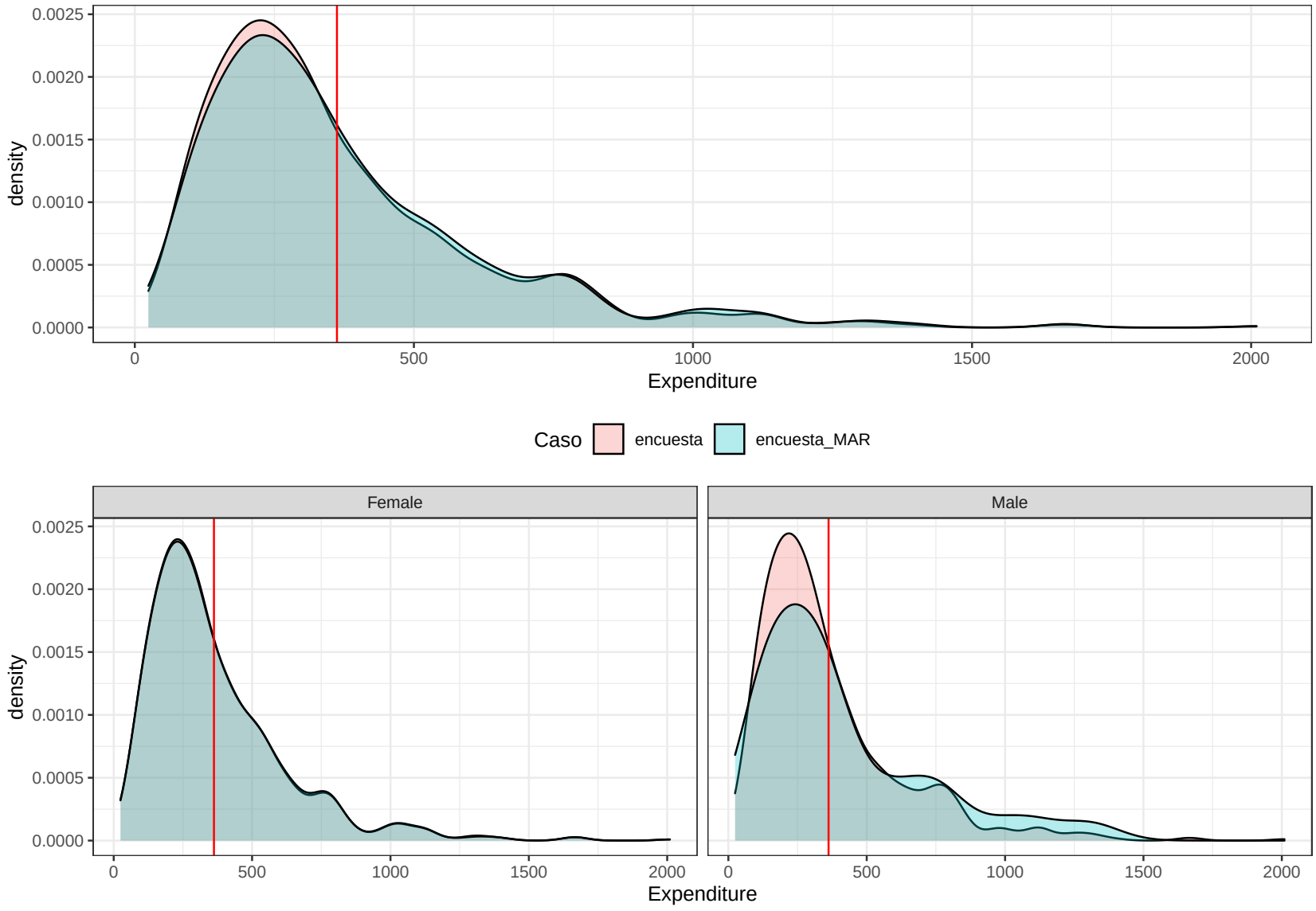


Creando valores perdidos

```
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



Creando valores perdidos

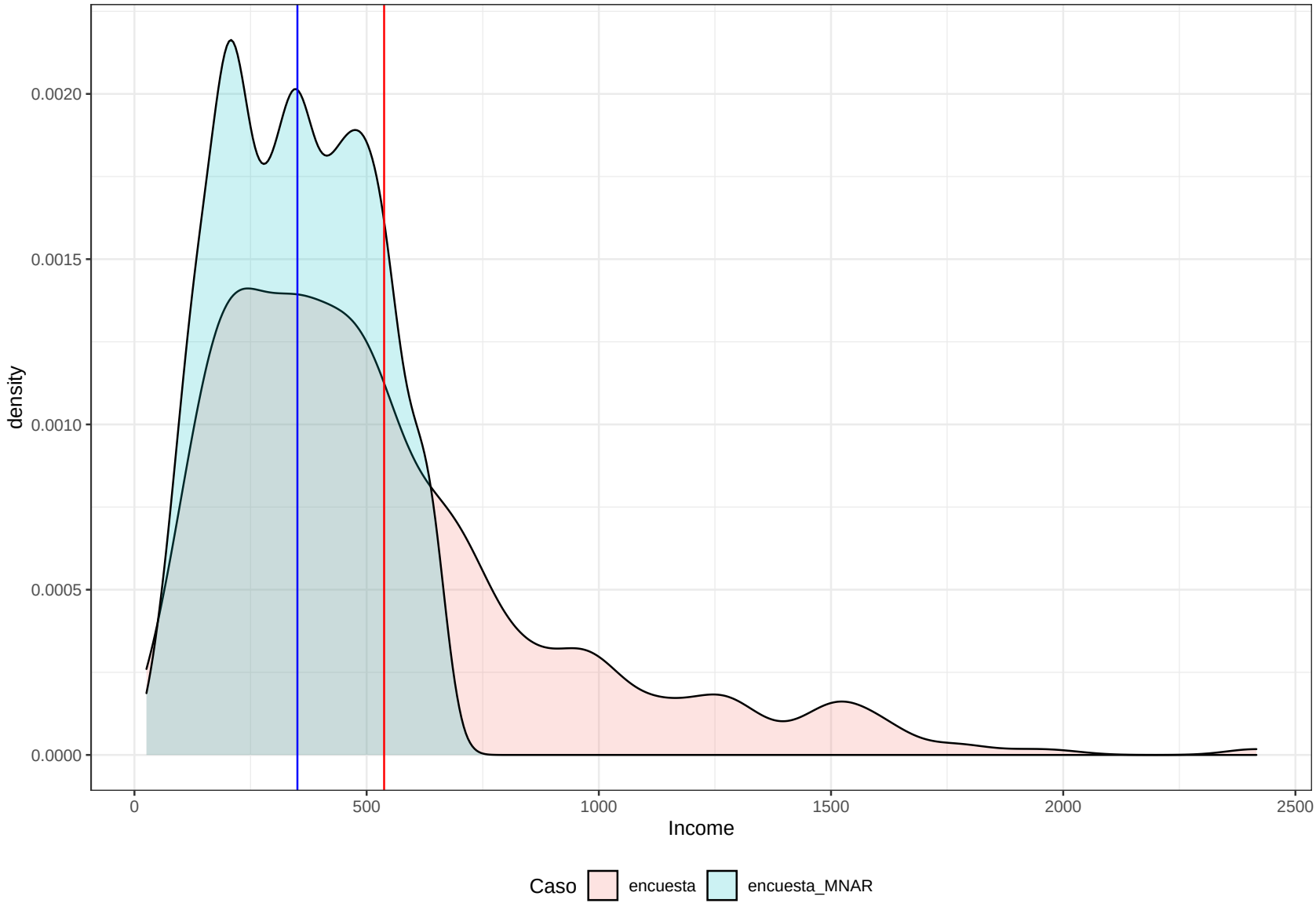
```
encuesta_MNAR <- encuesta %>%
  arrange((Income)) %>%
  slice(1:1300L)

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
```

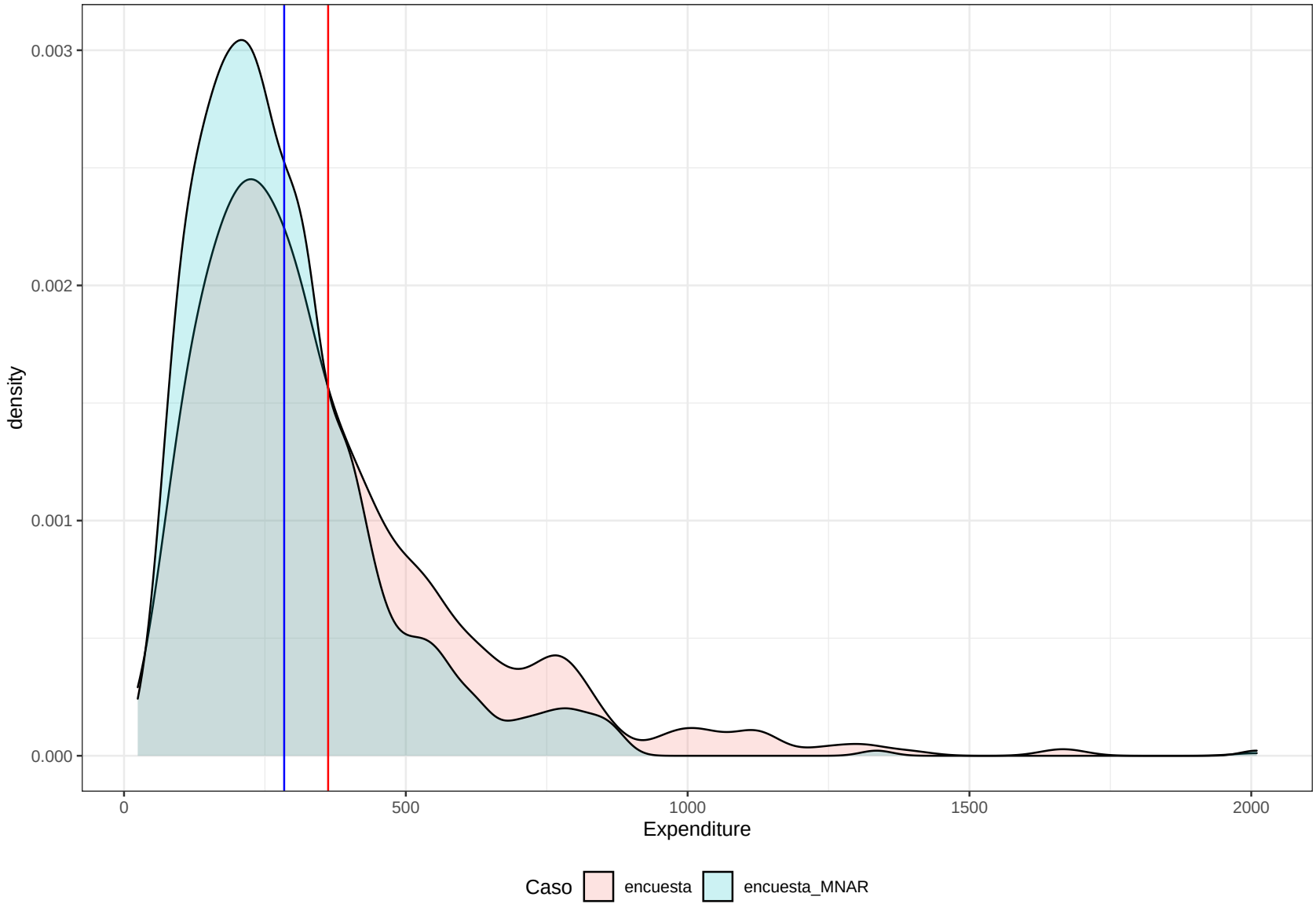
Creando valores perdidos



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

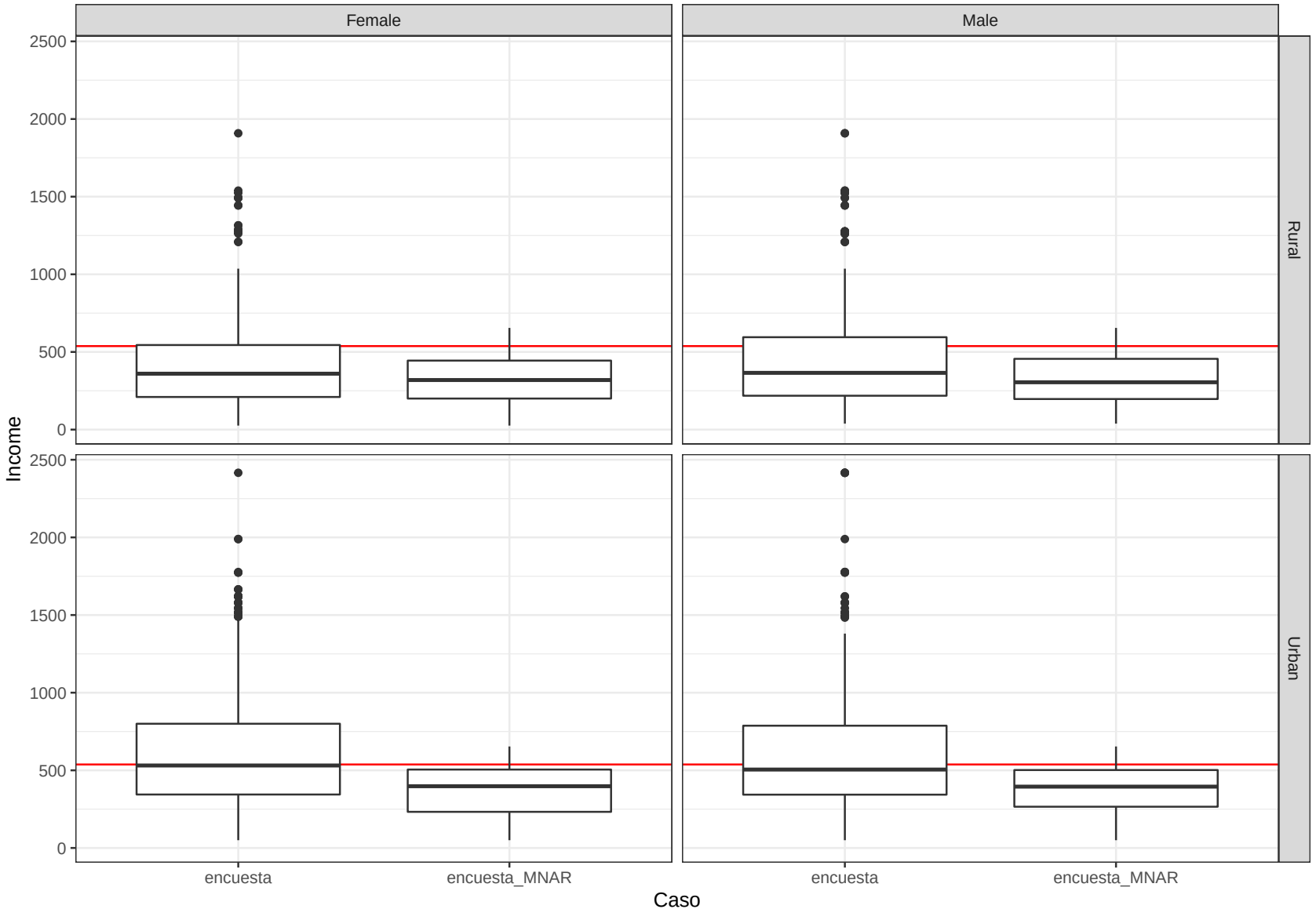


Creando valores perdidos

```
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



Creando valores perdidos

```
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.

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

Zone	Income
Rural	0.1977
Urban	0.2022

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

Sex	Income
Female	0.1920
Male	0.2097

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.

Imputación por la media no condicional.

```
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))
```

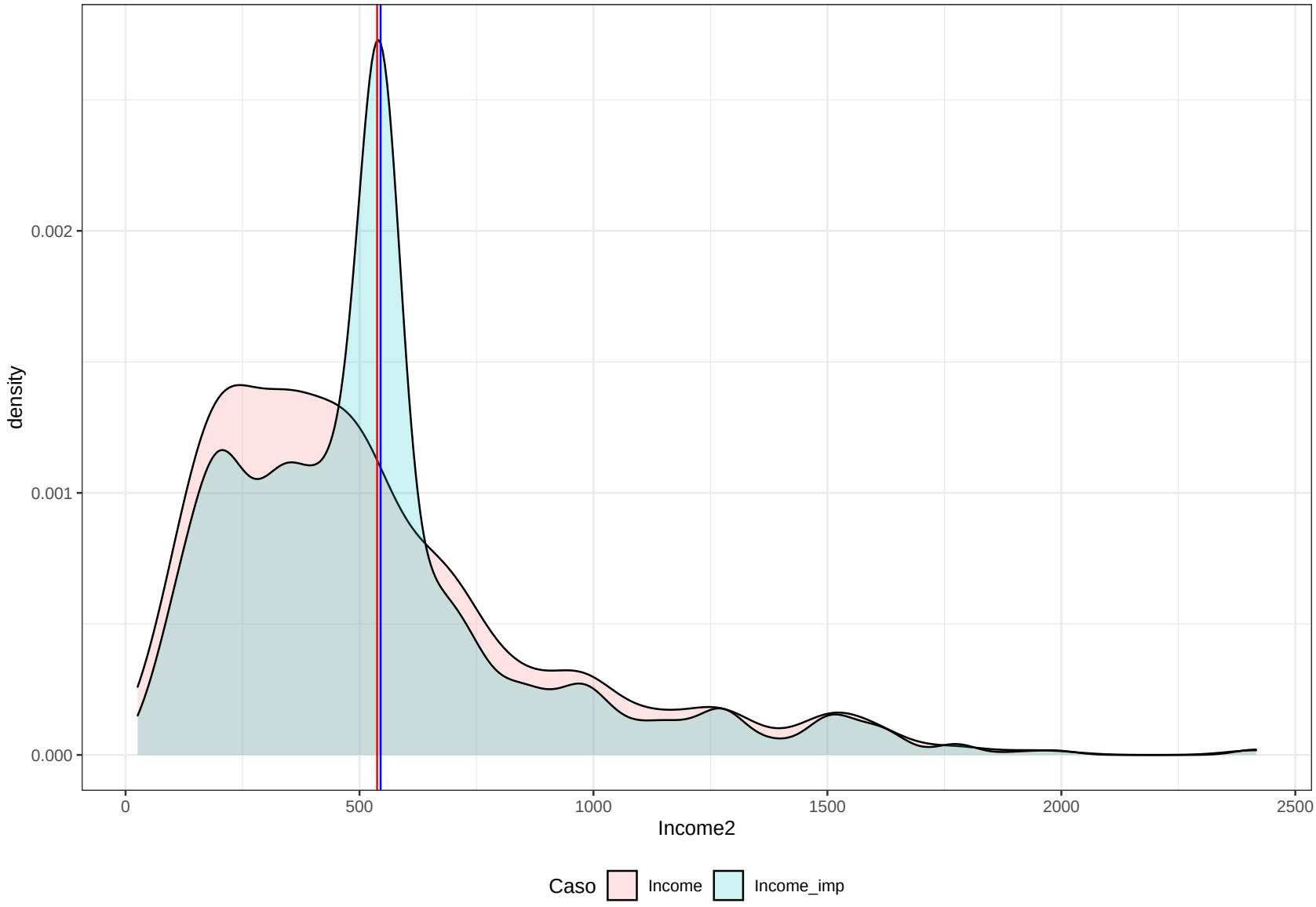
```
## [1] 0
```

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.



Imputación por la media no condicional.

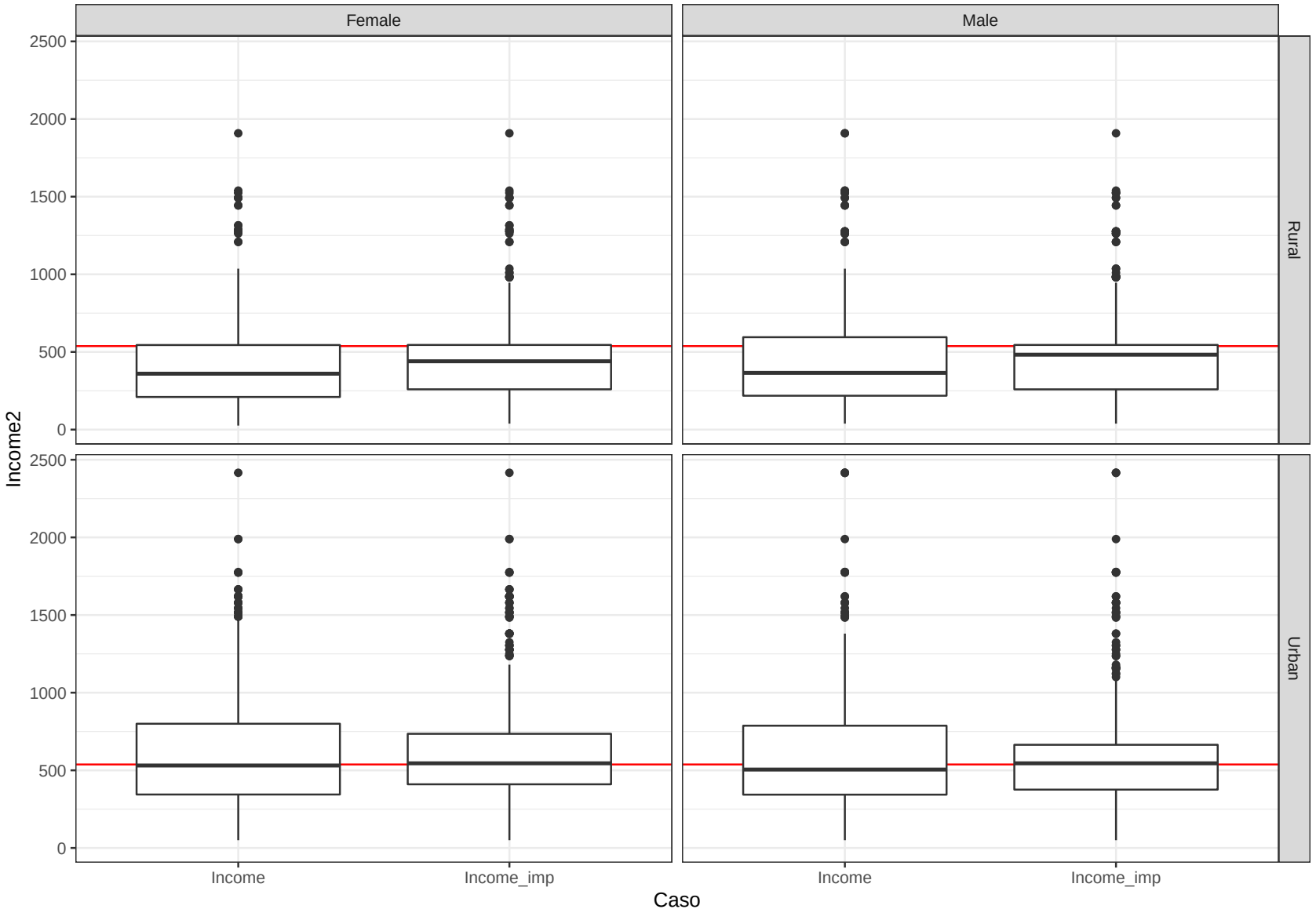
```
p1 <- ggplot(dat_plot4, aes(x= Caso, y = Income2)) +
   geom_hline(yintercept = mean(encuesta$Income),
                col = "red") +  geom_boxplot() +
  facet_grid(Zone~Sex) + theme_bw()

p1
```

Imputación por la media condicional.

Una variante del procedimiento anterior consiste en formar categorías a partir de covariables correlacionadas con la variable de interés, e imputar los datos omitidos con observaciones provenientes de la submuestra que comparte características comunes

Imputación por la media 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.

```
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
537.6	380.9	541.8	361.4

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))
```

Zone	Income_	Income_sd	Income_imp_	Income_imp_sd
Rural	443.6	314.0	448.6	293.4
Urban	629.6	416.6	632.9	396.8

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))
```

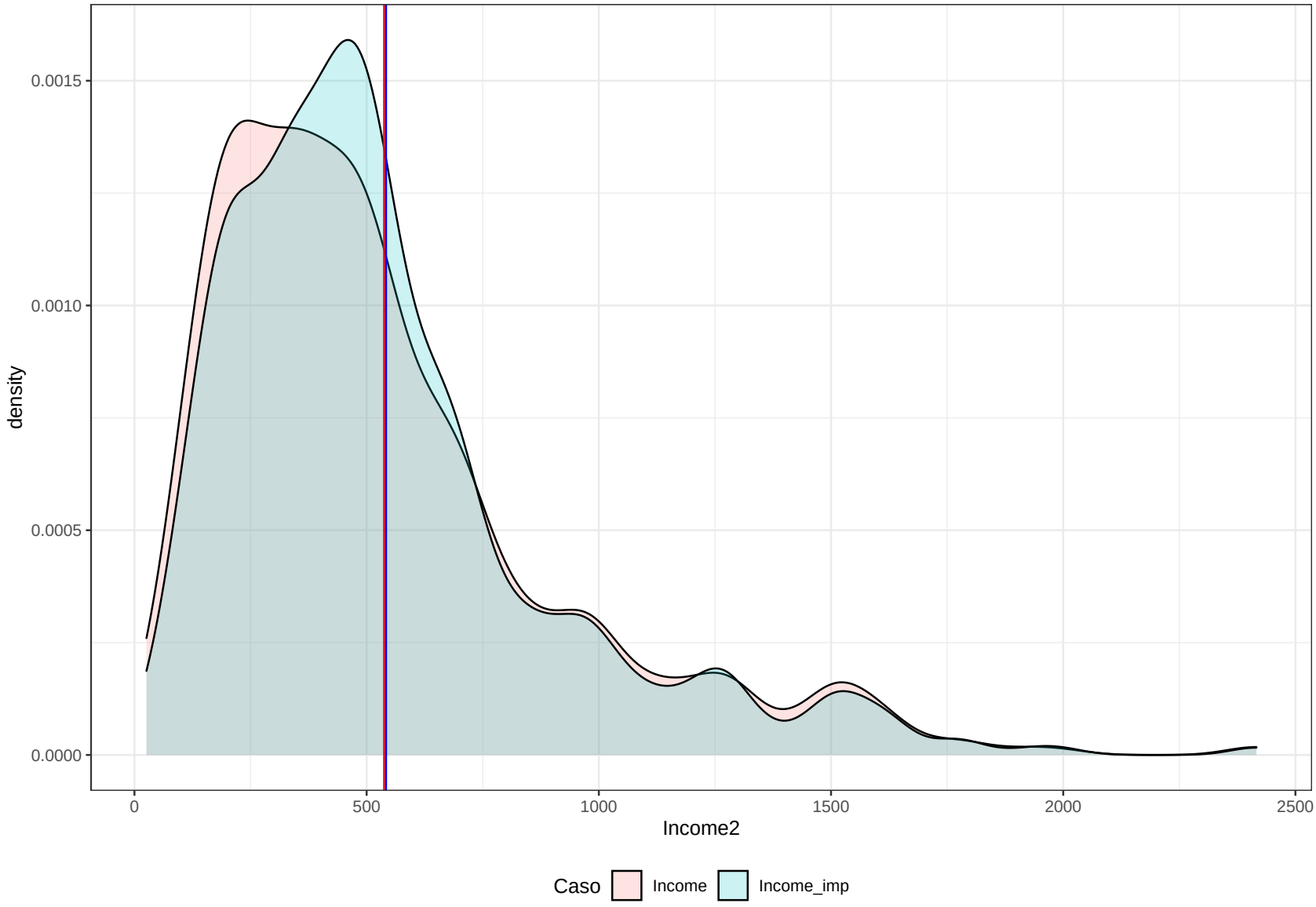
Sex	Income_	Income_sd	Income_imp_	Income_imp_sd
Female	534.7	374.1	540.9	357.7
Male	541.2	389.1	542.9	366.1

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
```

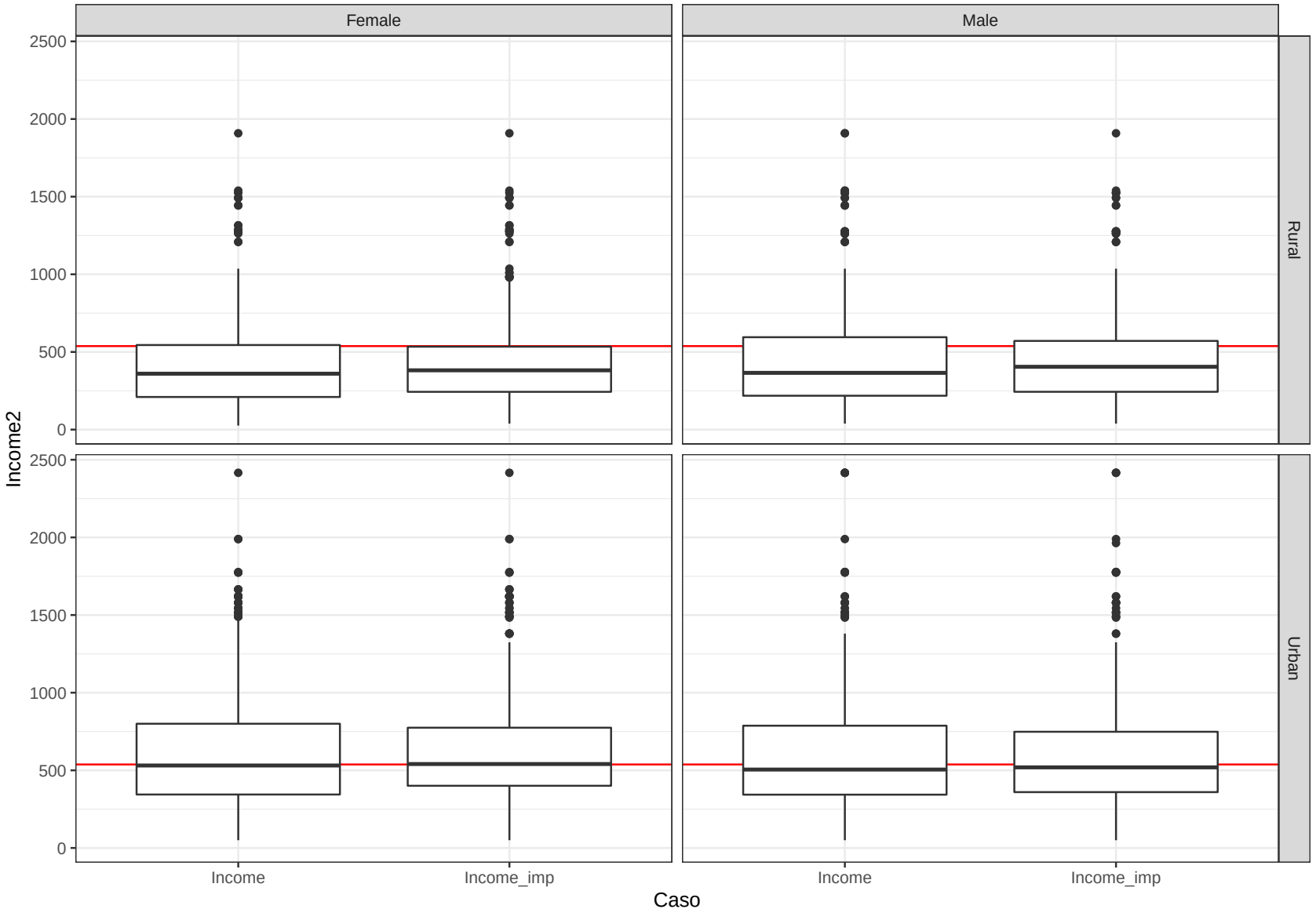
Imputación por la media condicional.



Imputación por la media condicional.

```
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.



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

```
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
537.6	380.9	546.3	394

Imputación por hot-deck

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

Zone	Income_	Income_sd	Income_imp_	Income_imp_sd
Rural	443.6	314.0	470.6	343.8
Urban	629.6	416.6	620.4	424.9

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))
```

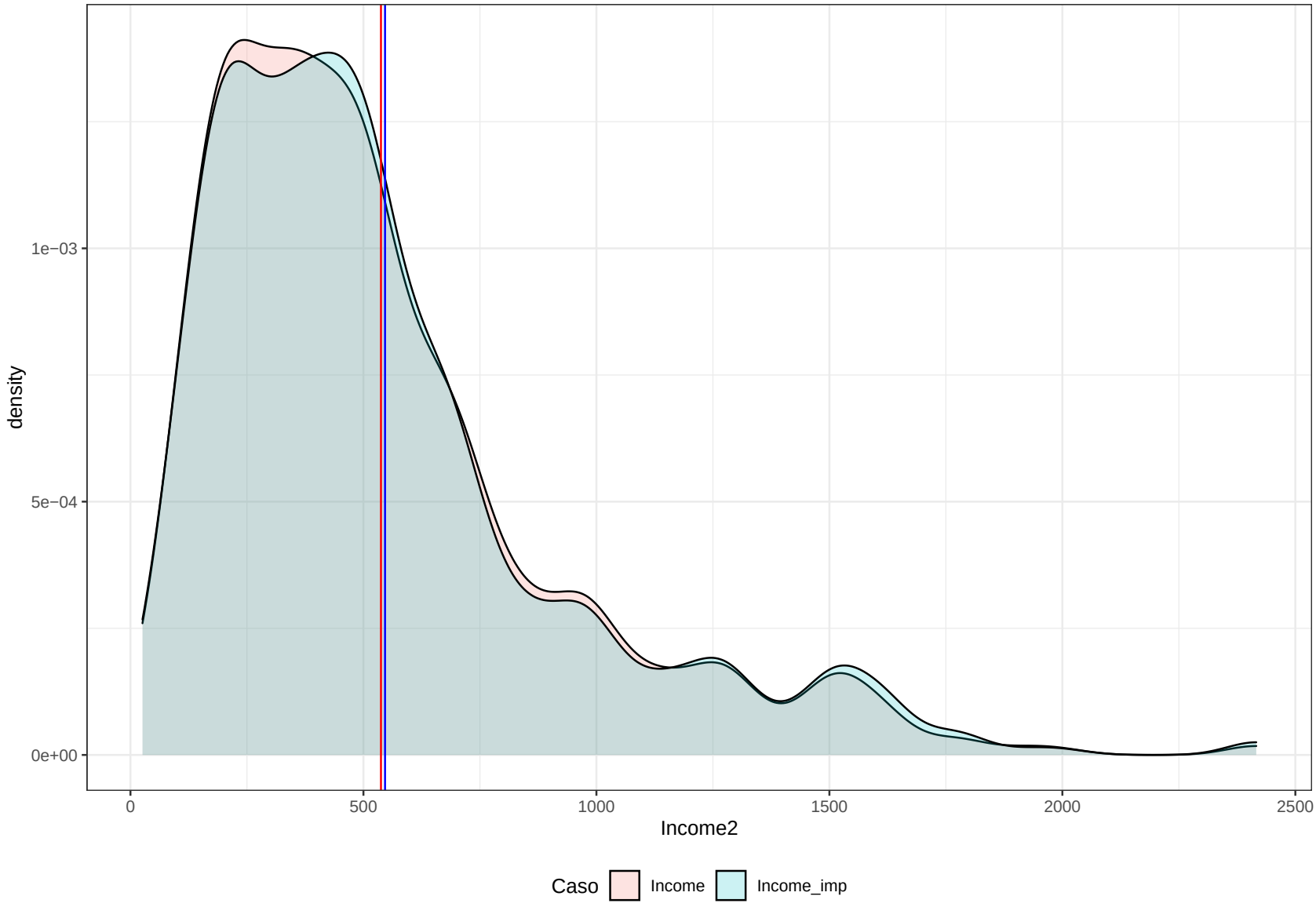
Sex	Income_	Income_sd	Income_imp_	Income_imp_sd
Female	534.7	374.1	548.3	388.8
Male	541.2	389.1	544.0	400.6

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
```

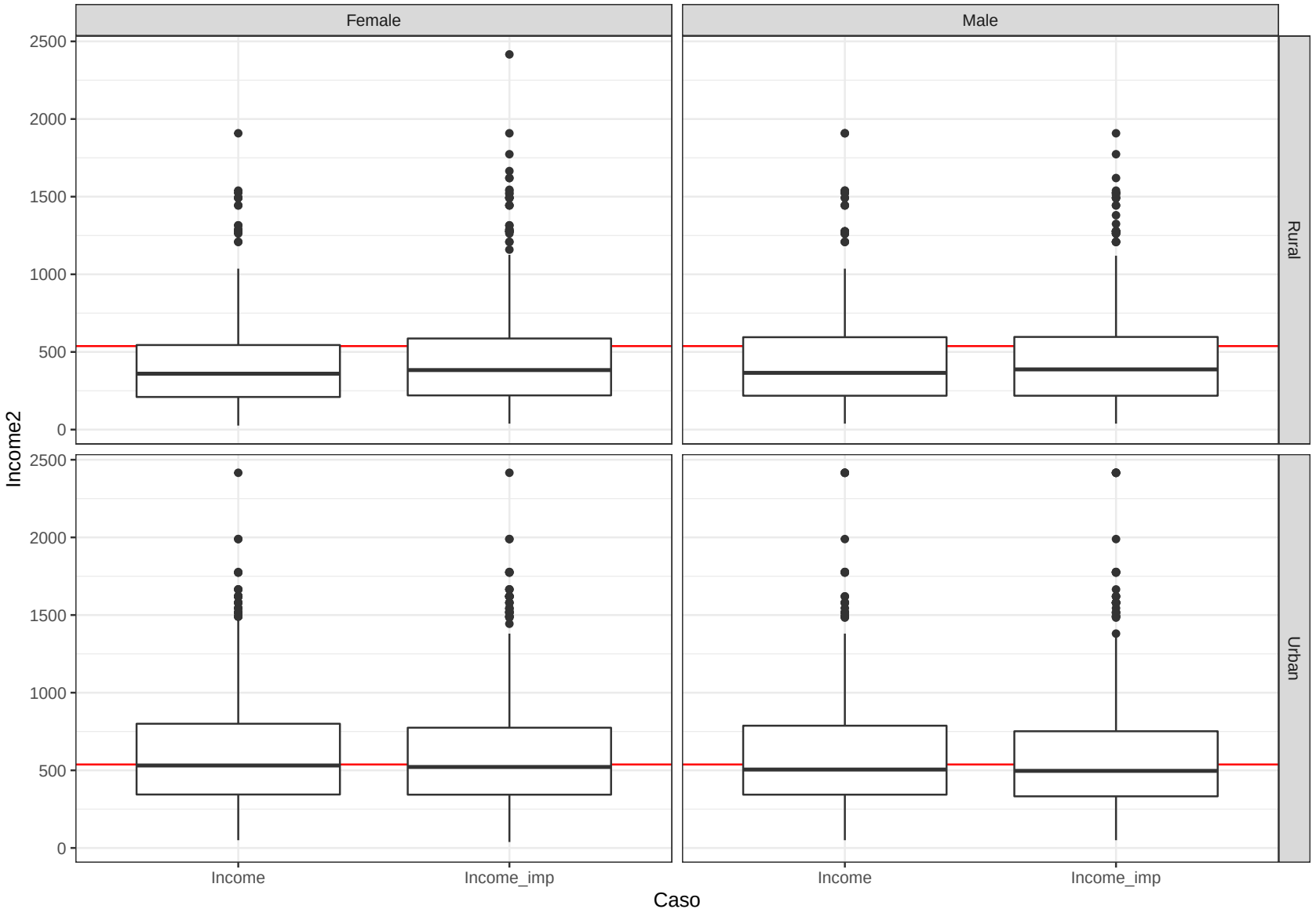
Imputación por hot-deck



Imputación por hot-deck

```
p1 <- ggplot(dat_plot6, 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.



Imputación por hot-deck

```
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
0.0471	0.3968	0.5562

```
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

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

Unemployed	Inactive	Employed	NA
0.0376	0.3174	0.4449	0.2

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

Unemployed	Inactive	Employed	NA
0.0489	0.3899	0.5612	0

Imputación por hot-deck

```
prop.table( table(encuesta$Zone, encuesta$Employment_missin
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Rural	0.0185	0.1618	0.2163	0.0978	0.4944
Urban	0.0191	0.1556	0.2287	0.1022	0.5056
NA	0.0000	0.0000	0.0000	0.0000	0.0000
Sum	0.0376	0.3174	0.4449	0.2000	1.0000

```
prop.table( table(encuesta$Zone, encuesta$Employment_imp,
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Rural	0.0230	0.1989	0.2725	0	0.4944
Urban	0.0258	0.1910	0.2888	0	0.5056
NA	0.0000	0.0000	0.0000	0	0.0000
Sum	0.0489	0.3899	0.5612	0	1.0000

Imputación por hot-deck

```
prop.table( table(encuesta$Sex, encuesta$Employment_missin
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Female	0.0112	0.2455	0.1876	0.1056	0.55
Male	0.0264	0.0719	0.2573	0.0944	0.45
NA	0.0000	0.0000	0.0000	0.0000	0.00
Sum	0.0376	0.3174	0.4449	0.2000	1.00

```
prop.table( table(encuesta$Sex, encuesta$Employment_imp,
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Female	0.0157	0.2860	0.2483	0	0.55
Male	0.0331	0.1039	0.3129	0	0.45
NA	0.0000	0.0000	0.0000	0	0.00
Sum	0.0489	0.3899	0.5612	0	1.00

Imputación por regresión

Se ajusta un modelo lineal que describa a y, variable a imputar, para un conjunto X de variables auxiliares que se deben disponer. Resuelve el problema de la distorsión de la distribución de la variable a imputar, pero puede crear inconsistencias dentro de la base de datos, pues podría obtenerse valores "imposibles", ya que el valor y es obtenido de variables auxiliares.

Imputación por regresión

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

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

```
## # weights:  15 (8 variable)
## initial  value 1564.423899
## iter  10 value 1099.747740
## final  value 1090.861650
## converged
```

Imputación por regresión

```
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

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

Unemployed	Inactive	Employed	NA
0.0376	0.3174	0.4449	0.2

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

Unemployed	Inactive	Employed	NA
0.0376	0.4169	0.5455	0

Imputación por regresión

```
prop.table( table(encuesta$Zone, encuesta$Employment_missin
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Rural	0.0185	0.1618	0.2163	0.0978	0.4944
Urban	0.0191	0.1556	0.2287	0.1022	0.5056
NA	0.0000	0.0000	0.0000	0.0000	0.0000
Sum	0.0376	0.3174	0.4449	0.2000	1.0000

```
prop.table( table(encuesta$Zone, encuesta$Employment_imp,
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Rural	0.0185	0.2118	0.2640	0	0.4944
Urban	0.0191	0.2051	0.2815	0	0.5056
NA	0.0000	0.0000	0.0000	0	0.0000
Sum	0.0376	0.4169	0.5455	0	1.0000

Imputación por regresión

```
prop.table( table(encuesta$Sex, encuesta$Employment_missin
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Female	0.0112	0.2455	0.1876	0.1056	0.55
Male	0.0264	0.0719	0.2573	0.0944	0.45
NA	0.0000	0.0000	0.0000	0.0000	0.00
Sum	0.0376	0.3174	0.4449	0.2000	1.00

```
prop.table( table(encuesta$Sex, encuesta$Employment_imp,
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Female	0.0112	0.3449	0.1938	0	0.55
Male	0.0264	0.0719	0.3517	0	0.45
NA	0.0000	0.0000	0.0000	0	0.00
Sum	0.0376	0.4169	0.5455	0	1.00

Imputación por regresión

```
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
537.6	380.9	543.6	366.4

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))
```

Zone	Income_	Income_sd	Income_imp_	Income_imp_sd
Rural	443.6	314.0	446.5	299.3
Urban	629.6	416.6	638.4	399.9

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))
```

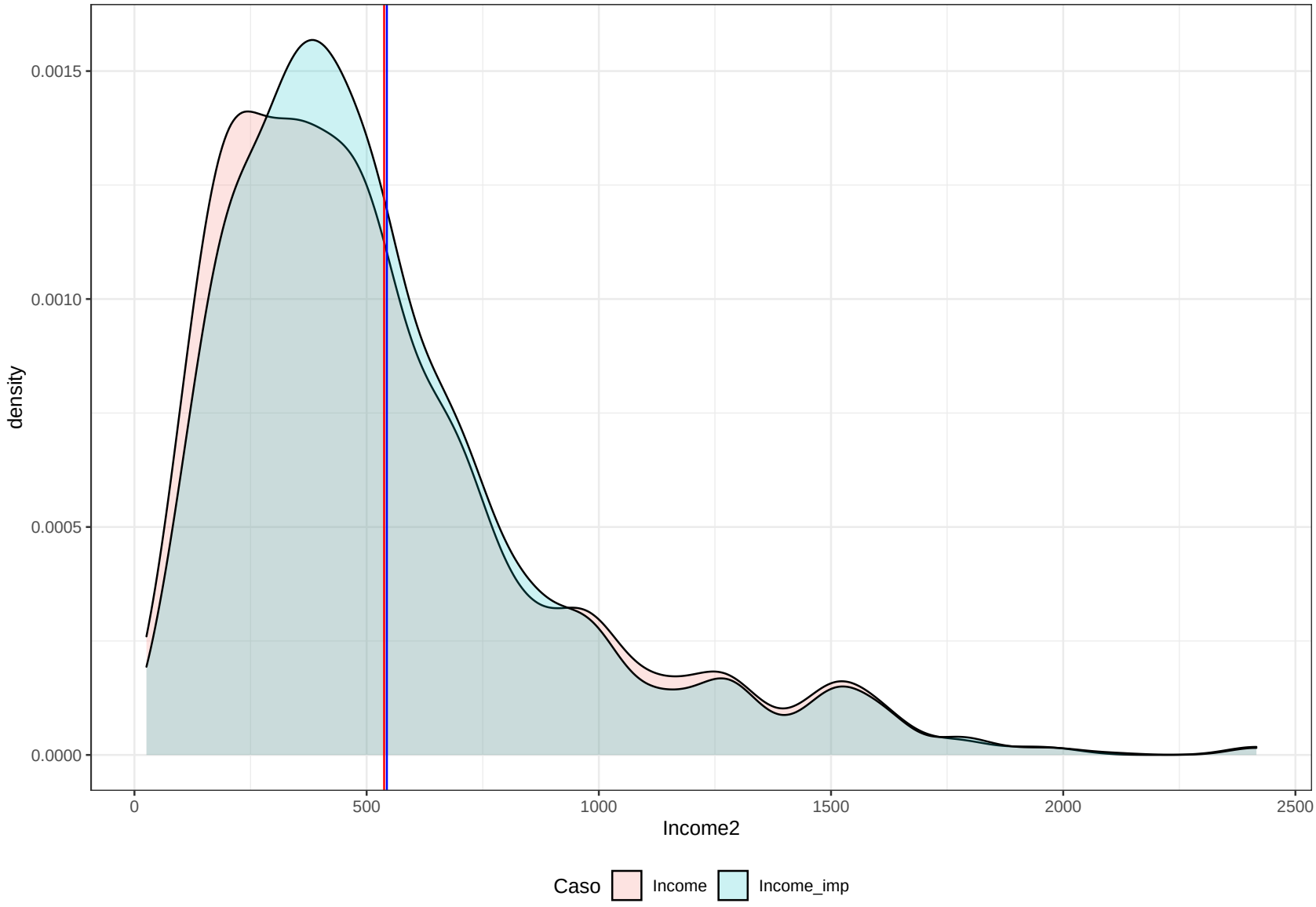
Sex	Income_	Income_sd	Income_imp_	Income_imp_sd
Female	534.7	374.1	548.0	367.1
Male	541.2	389.1	538.1	365.7

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
```

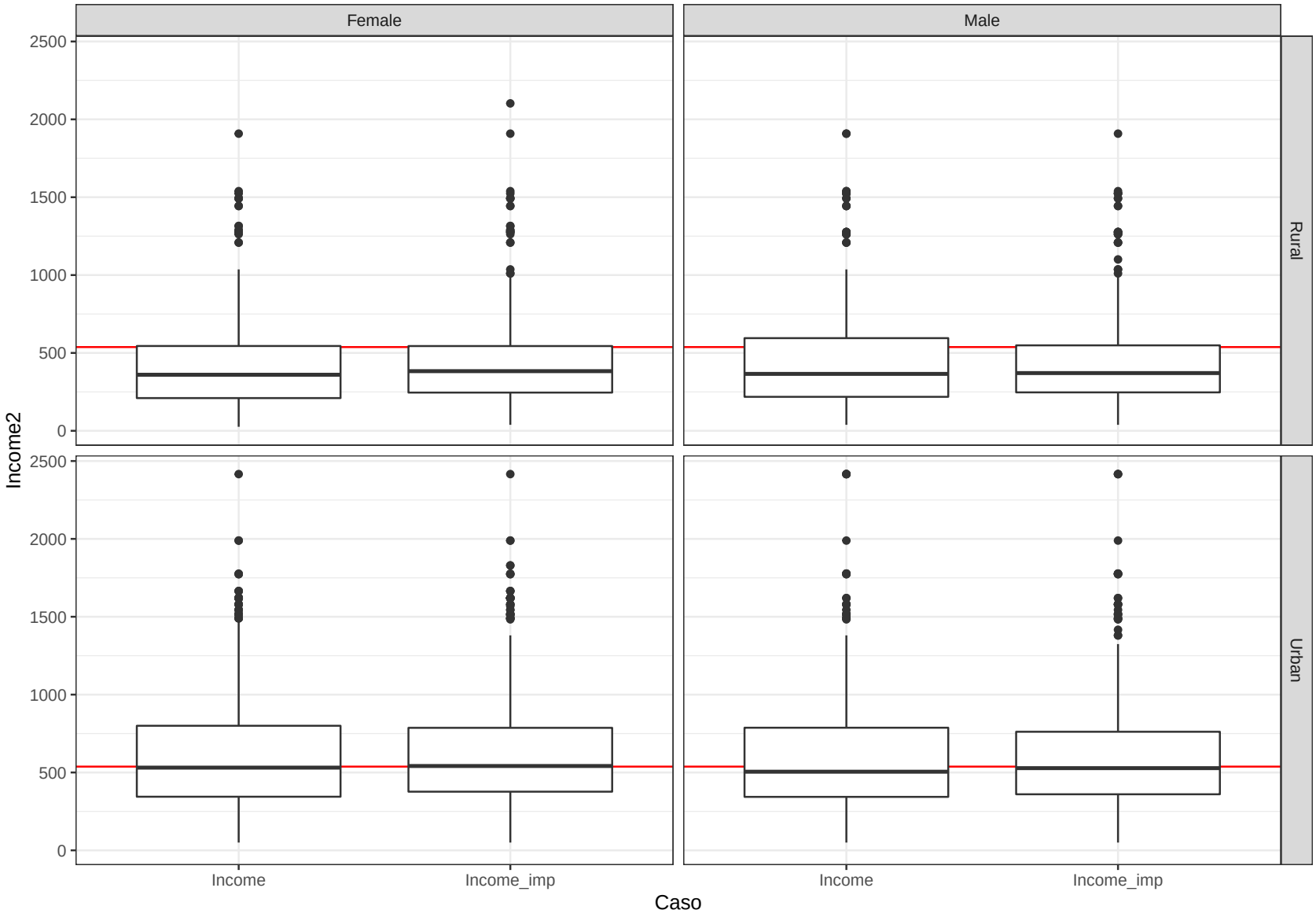
Imputación por regresión



Imputación por regresión

```
p1 <- ggplot(dat_plot7, aes(x= Caso, y = Income2)) +
   geom_hline(yintercept = mean(encuesta$Income),
              col = "red") +  geom_boxplot() +
  facet_grid(Zone~Sex) + theme_bw()
p1
```

Imputación por regresión



Imputación por el vecino más cercano

- **Paso 1**: Definir una magnitud de distancia (Distancia euclidiana, k-media, K-Medioides).
- **Paso 2**: Para la i-ésimo elemento identificar el donante, cual será el más cercano al receptor según la magnitud de distancia previamente definida.
- **Paso 3**: Se imputa el valor faltante con la información del donante identificado previamente.

Imputación por el vecino más cercano

```
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))
for(ii in 1:nrow(encuesta_no_obs)){
Expen_ii <- encuesta_no_obs$Expenditure[[ii]]
don_ii <- which.min(abs(Expen_ii -
                          encuesta_obs$Expenditure))
encuesta_no_obs$Income_imp[[ii]] <-
  encuesta_obs$Income_missin[[don_ii]]
encuesta_no_obs$Employment_imp[[ii]] <-
  encuesta_obs$Employment_missin[[don_ii]]
}

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

Imputación por el vecino más cercano

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

Unemployed	Inactive	Employed	NA
0.0376	0.3174	0.4449	0.2

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

Unemployed	Inactive	Employed	NA
0.0489	0.3882	0.5629	0

Imputación por el vecino más cercano

```
prop.table( table(encuesta$Zone, encuesta$Employment_missin
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Rural	0.0185	0.1618	0.2163	0.0978	0.4944
Urban	0.0191	0.1556	0.2287	0.1022	0.5056
NA	0.0000	0.0000	0.0000	0.0000	0.0000
Sum	0.0376	0.3174	0.4449	0.2000	1.0000

```
prop.table( table(encuesta$Zone, encuesta$Employment_imp,
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Rural	0.0230	0.2034	0.2680	0	0.4944
Urban	0.0258	0.1848	0.2949	0	0.5056
NA	0.0000	0.0000	0.0000	0	0.0000
Sum	0.0489	0.3882	0.5629	0	1.0000

Imputación por el vecino más cercano

```
prop.table( table(encuesta$Sex, encuesta$Employment_missin
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Female	0.0112	0.2455	0.1876	0.1056	0.55
Male	0.0264	0.0719	0.2573	0.0944	0.45
NA	0.0000	0.0000	0.0000	0.0000	0.00
Sum	0.0376	0.3174	0.4449	0.2000	1.00

```
prop.table( table(encuesta$Sex, encuesta$Employment_imp,
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Female	0.0191	0.2725	0.2584	0	0.55
Male	0.0298	0.1157	0.3045	0	0.45
NA	0.0000	0.0000	0.0000	0	0.00
Sum	0.0489	0.3882	0.5629	0	1.00

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))
```

Income_	Income_sd	Income_imp_	Income_imp_sd
537.6	380.9	541.3	380.8

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))
```

Zone	Income_	Income_sd	Income_imp_	Income_imp_sd
Rural	443.6	314.0	447.5	314.8
Urban	629.6	416.6	633.0	416.0

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))
```

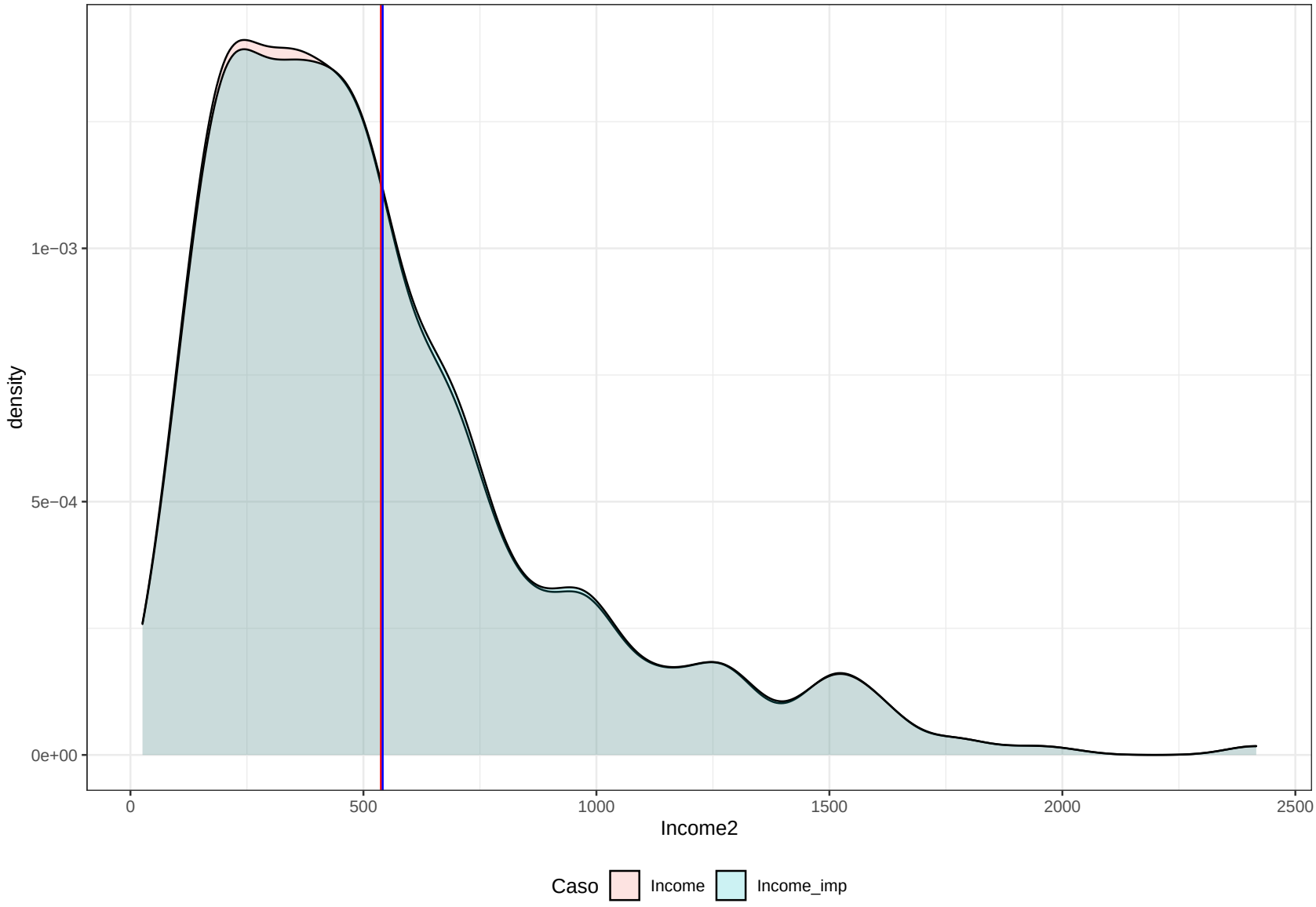
Sex	Income_	Income_sd	Income_imp_	Income_imp_sd
Female	534.7	374.1	538.8	373.2
Male	541.2	389.1	544.4	390.2

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
```

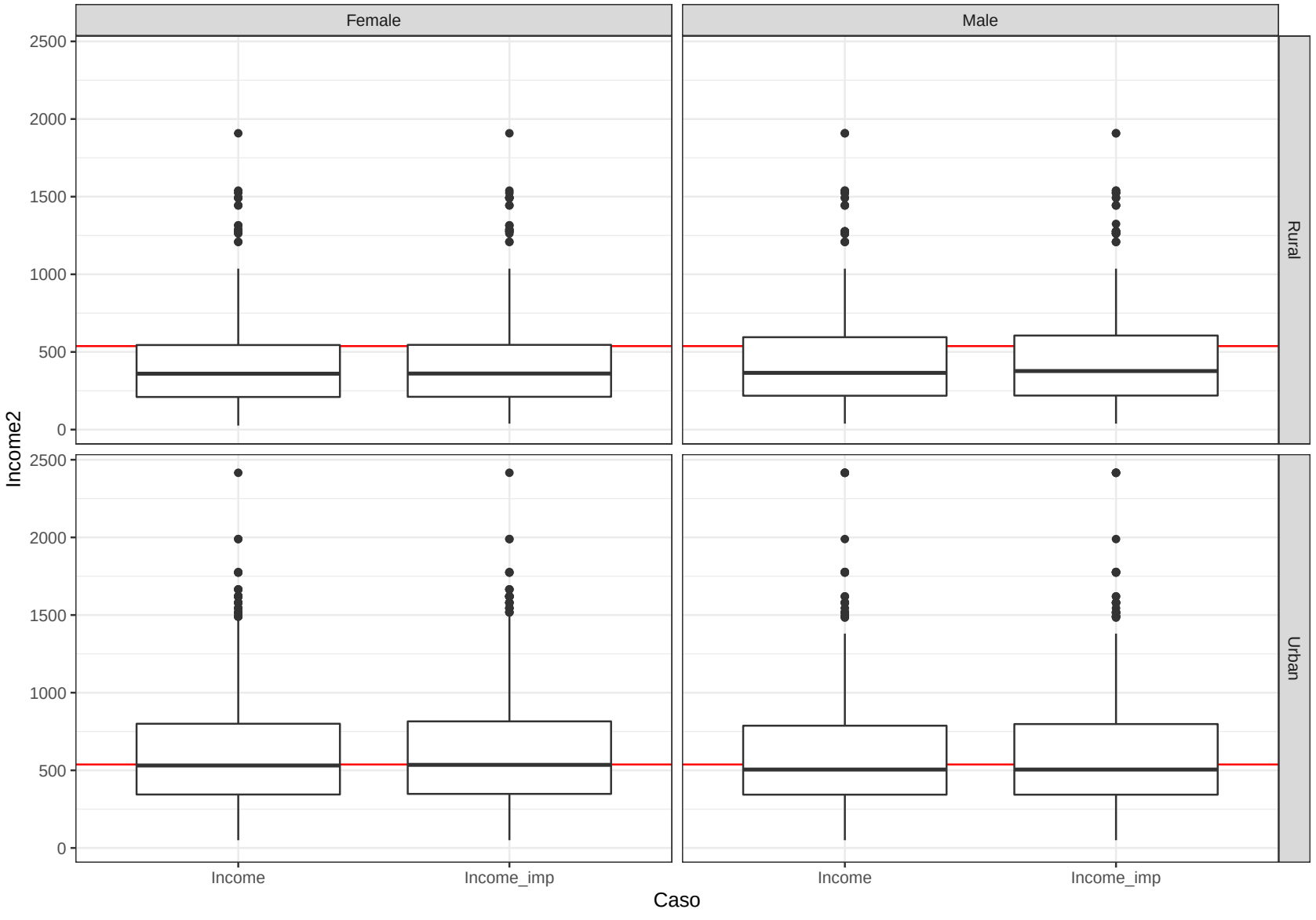
Imputación por el vecino más cercano



Imputación por el vecino más cercano

```
p1 <- ggplot(dat_plot8, aes(x= Caso, y = Income2)) +
   geom_hline(yintercept = mean(encuesta$Income),
              col = "red") +  geom_boxplot() +
  facet_grid(Zone~Sex) + theme_bw()
p1
```

Imputación por el vecino más cercano



Imputación por el vecino más cercano con 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

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

Imputación por el vecino más cercano con regresión

```
pred_Obs <- predict(mod, encuesta_obs)
pred_no_Obs <- predict(mod, encuesta_no_obs)

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]]
}

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

Imputación por el vecino más cercano con regresión

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

Unemployed	Inactive	Employed	NA
0.0376	0.3174	0.4449	0.2

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

Unemployed	Inactive	Employed	NA
0.0483	0.3899	0.5618	0

Imputación por el vecino más cercano con regresión

```
prop.table( table(encuesta$Zone, encuesta$Employment_missi
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Rural	0.0185	0.1618	0.2163	0.0978	0.4944
Urban	0.0191	0.1556	0.2287	0.1022	0.5056
NA	0.0000	0.0000	0.0000	0.0000	0.0000
Sum	0.0376	0.3174	0.4449	0.2000	1.0000

```
prop.table( table(encuesta$Zone, encuesta$Employment_imp,
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Rural	0.0242	0.1994	0.2708	0	0.4944
Urban	0.0242	0.1904	0.2910	0	0.5056
NA	0.0000	0.0000	0.0000	0	0.0000
Sum	0.0483	0.3899	0.5618	0	1.0000

Imputación por el vecino más cercano con regresión

```
prop.table( table(encuesta$Sex, encuesta$Employment_missin
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Female	0.0112	0.2455	0.1876	0.1056	0.55
Male	0.0264	0.0719	0.2573	0.0944	0.45
NA	0.0000	0.0000	0.0000	0.0000	0.00
Sum	0.0376	0.3174	0.4449	0.2000	1.00

```
prop.table( table(encuesta$Sex, encuesta$Employment_imp,
        useNA = "a")) %>% addmargins()
```

/	Unemployed	Inactive	Employed	NA	Sum
Female	0.0157	0.2882	0.2461	0	0.55
Male	0.0326	0.1017	0.3157	0	0.45
NA	0.0000	0.0000	0.0000	0	0.00
Sum	0.0483	0.3899	0.5618	0	1.00

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))
```

Income_	Income_sd	Income_imp_	Income_imp_sd
537.6	380.9	542.4	382.4

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))
```

Zone	Income_	Income_sd	Income_imp_	Income_imp_sd
Rural	443.6	314.0	446.0	315.3
Urban	629.6	416.6	636.7	417.5

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))
```

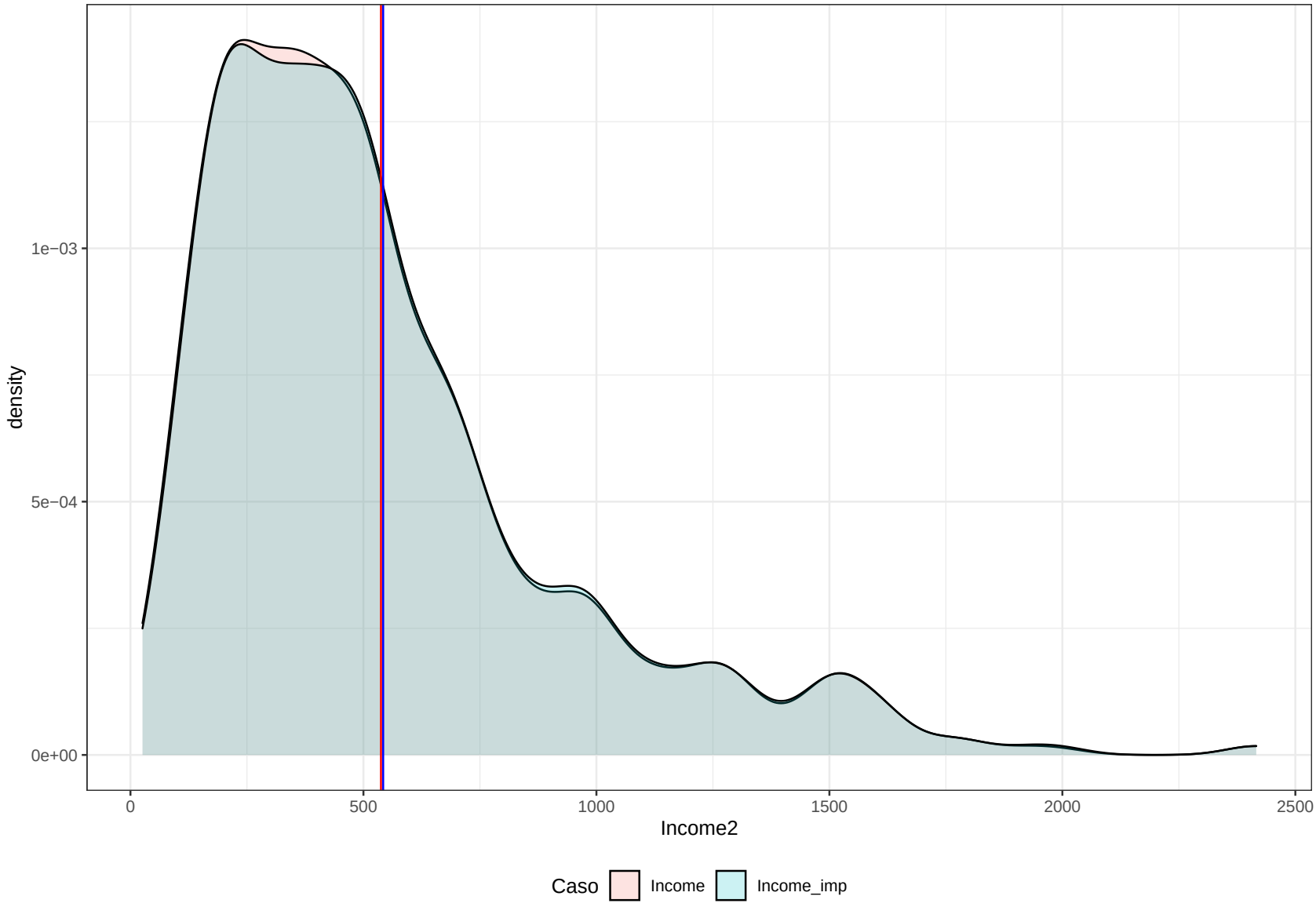
Sex	Income_	Income_sd	Income_imp_	Income_imp_sd
Female	534.7	374.1	539.1	374.5
Male	541.2	389.1	546.3	392.1

Imputación por el vecino más cercano con regresión

```
## 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
```

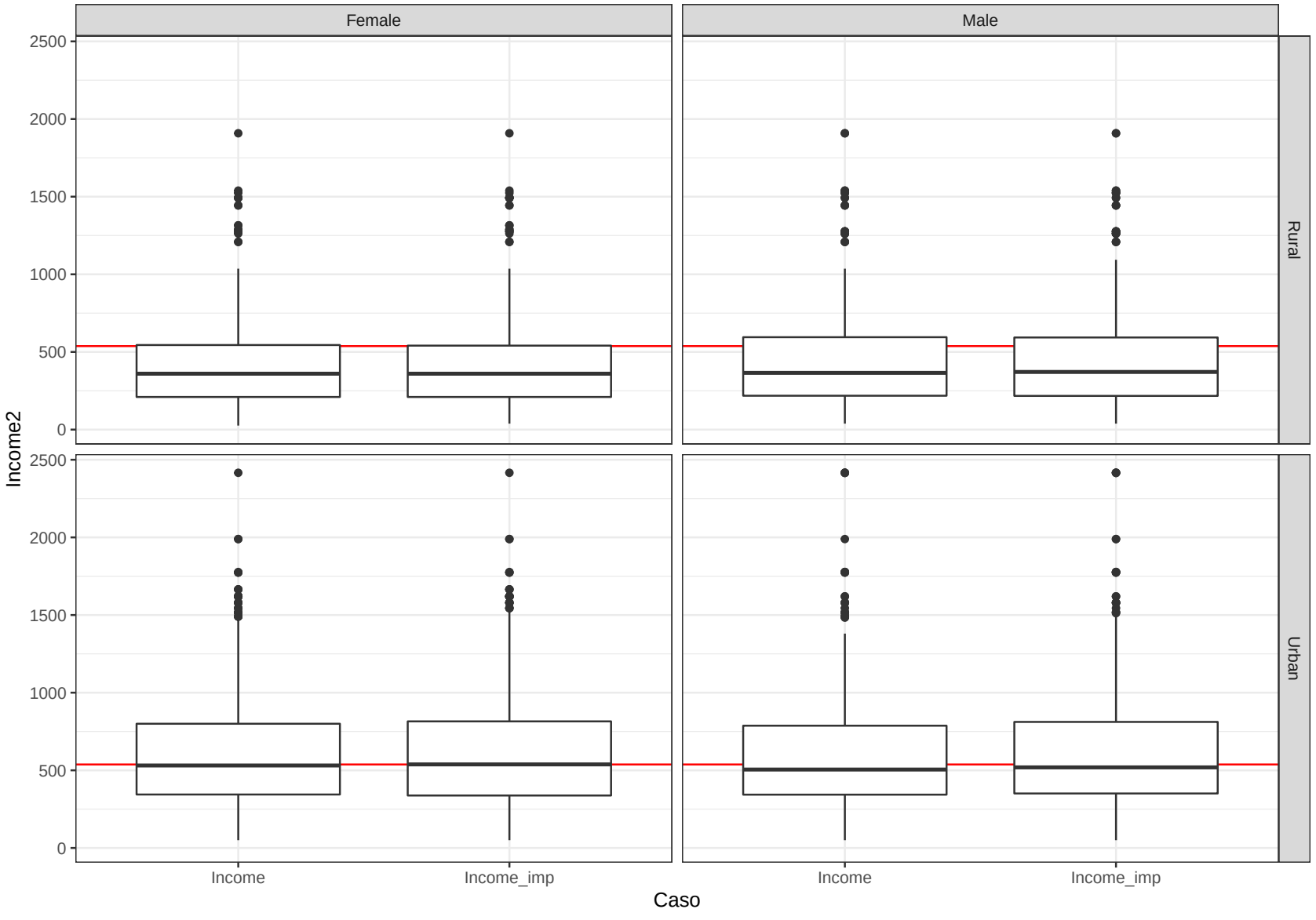
Imputación por el vecino más cercano



Imputación por el vecino más cercano

```
p1 <- ggplot(dat_plot9, aes(x= Caso, y = Income2)) +
   geom_hline(yintercept = mean(encuesta$Income),
              col = "red") +  geom_boxplot() +
  facet_grid(Zone~Sex) + theme_bw()
p1
```

Imputación por el vecino más cercano



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 $Var(\varepsilon) = \sigma^2$.

Introducción a la imputación múltiple.

- 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 $n = 500$ datos con una pendiente $\beta = 10$ y con una dispersión de $\sigma = 2$. A su vez, el conjunto de datos tendrá $n_0 = 200$ valores faltantes en la variable respuesta.

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)
```

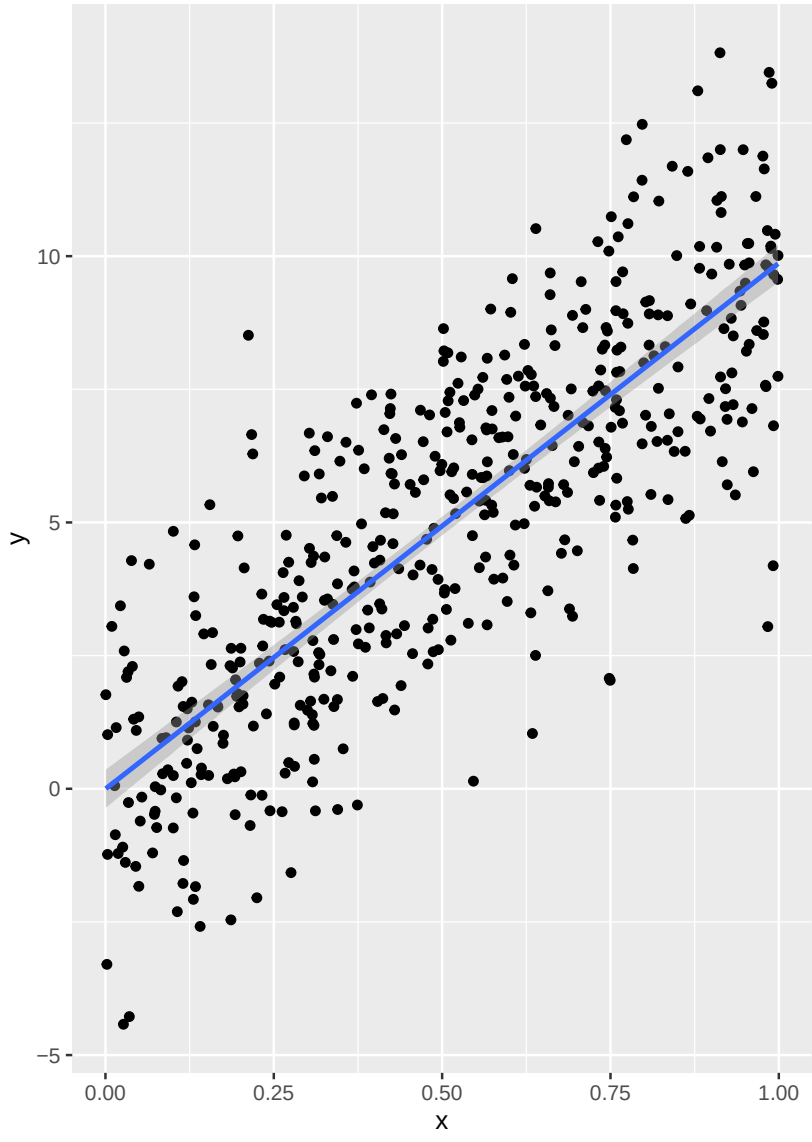
x	y	faltantes	y.per
0.1137	2.0109	No	2.011
0.6223	8.3432	No	8.343
0.6093	6.9971	No	6.997
0.6234	7.5602	Si	NA
0.8609	6.3364	No	6.336
0.6403	5.6621	No	5.662
0.0095	3.0489	No	3.049
0.2326	-0.1223	Si	NA
0.6661	7.1770	Si	NA
0.5143	5.9525	No	5.952
0.6936	8.8875	No	8.887
0.5450	4.7520	No	4.752

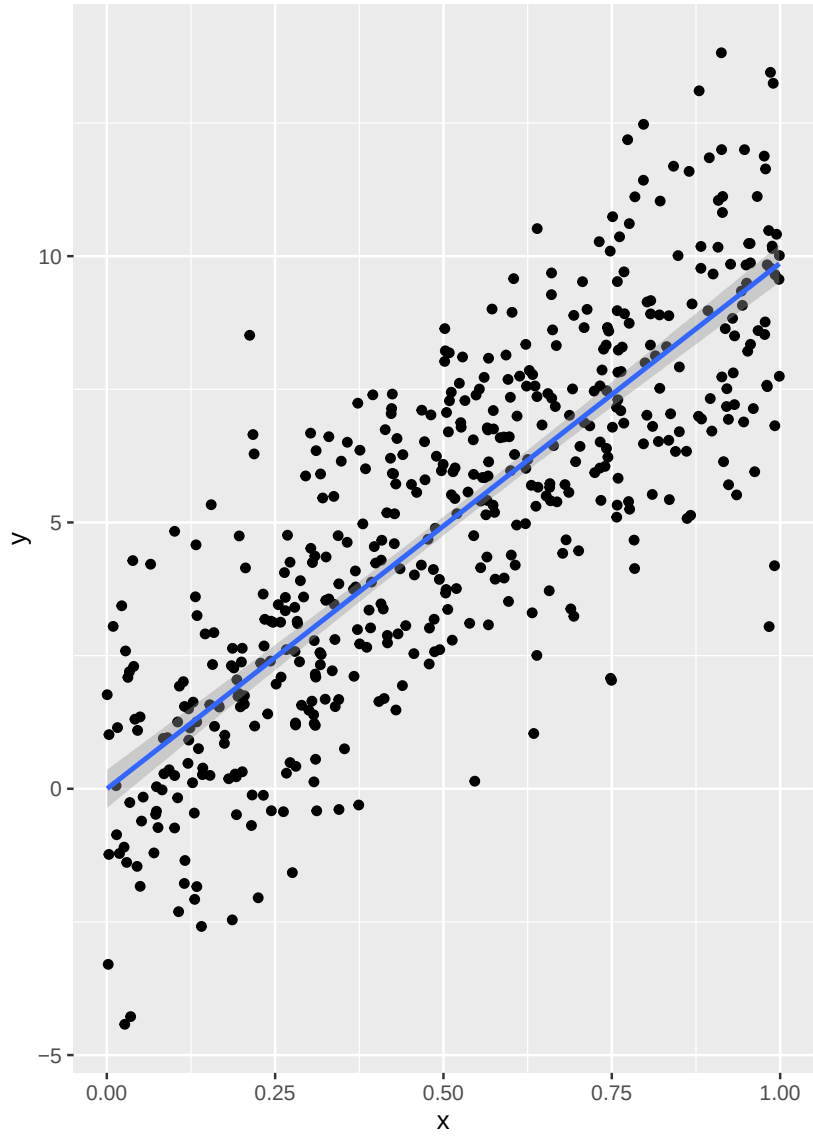
Introducción a la imputación múltiple.

```
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.





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.

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

Introducción a la imputación múltiple

Hay varias maneras de realizar la imputación:

- **Ingenua**: Esta clase de imputación carece de aleatoriedad y por tanto, la varianza de β va a ser subestimada.
- **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.
- **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.

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:

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

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

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

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:

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

```
## [1] 10.3
```

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

```
## [1] 0.2222
```

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

x	y	faltantes	y.per
0.2173	0.2872	Si	0.4423
0.2953	1.5861	Si	1.8694
0.9609	11.2310	Si	11.8596
0.3120	5.0509	No	5.0509
0.0521	-0.7714	Si	-0.5381

Imputación Bootstrap

Nótese que existe una buena dispersión en los valores imputados.

Imputación Bootstrap en 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.

```
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)
```

Imputación Bootstrap en la encuesta.

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

Income	Income_vp_1	Income_vp_2	Income_vp_3
243.2	349.2	378.9	361.7
223.0	235.3	245.0	247.1
223.0	221.3	264.3	237.0
337.5	364.6	360.9	373.2
337.5	350.6	380.1	363.1
224.3	240.3	249.5	251.9
464.2	429.4	418.9	436.4
464.2	415.4	438.2	426.3
260.0	443.1	440.6	429.4
1380.4	855.3	809.9	831.5

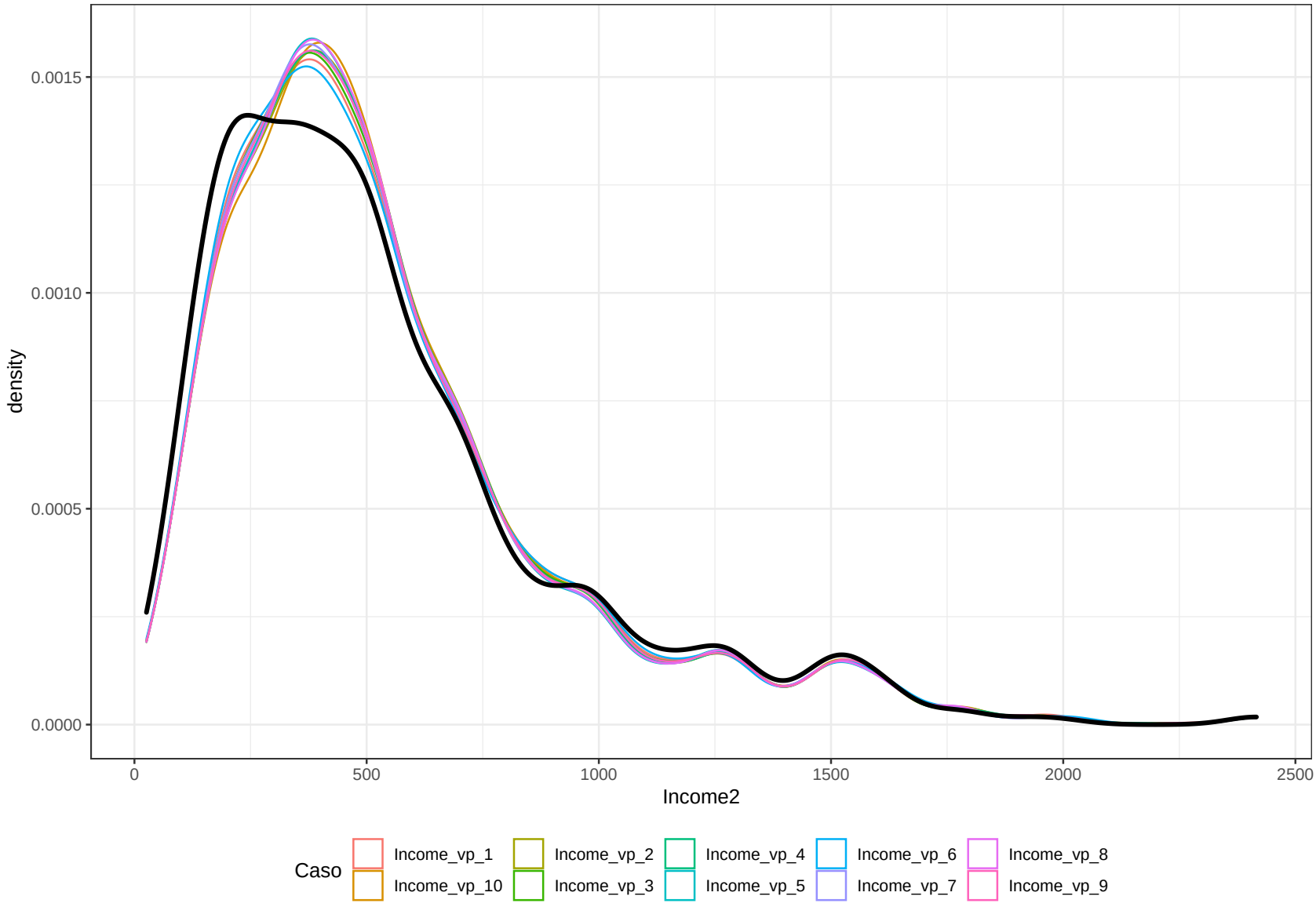
Imputación Bootstrap en la encuesta.

```
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

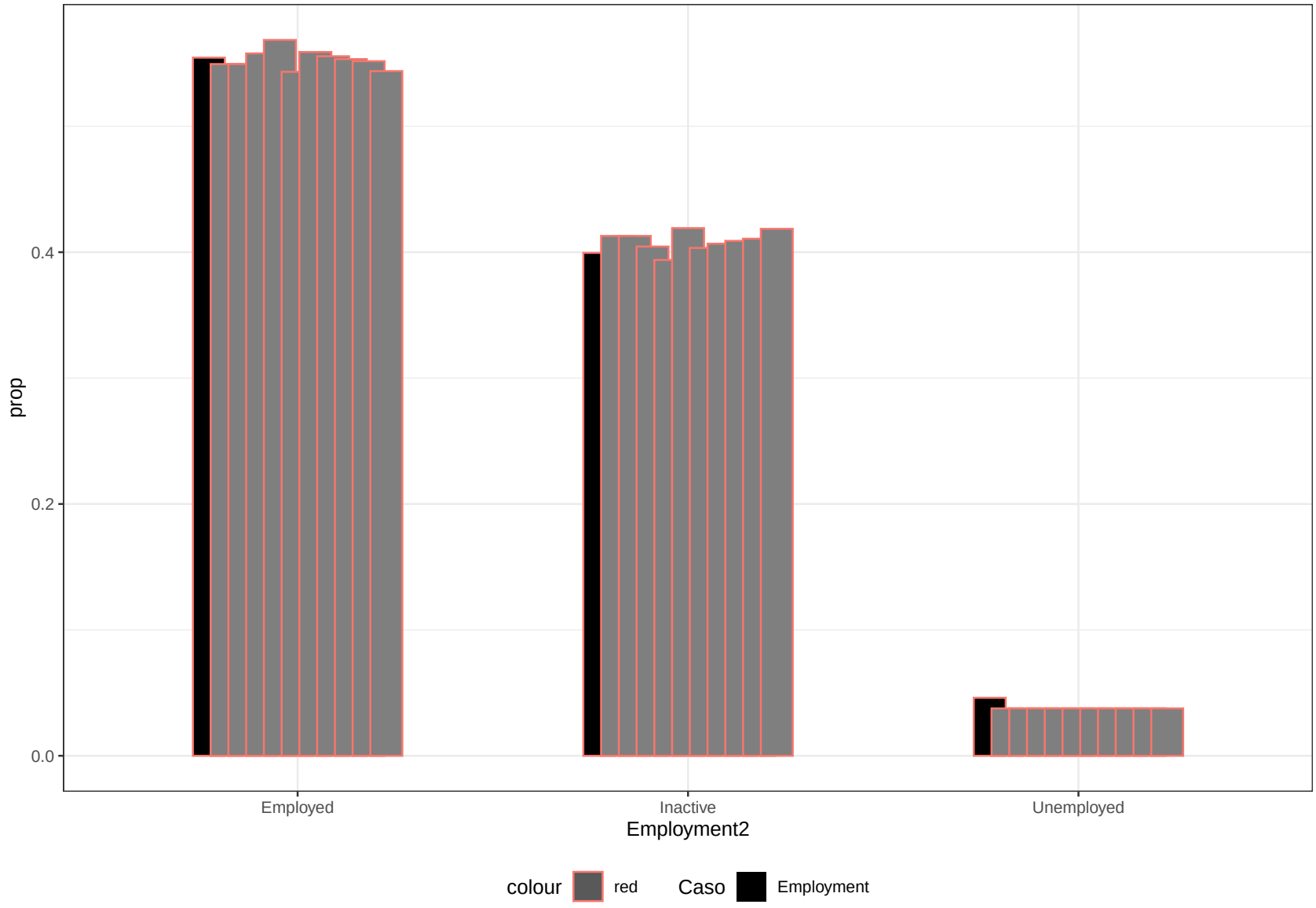


Imputación Bootstrap en la encuesta.

```
## 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



Definir diseño de la muestra con `srvyr`

```
library(srvyr)

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

Estimación del promedio con valores plausibles (vp)

```
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	565.7	547.4
2	566.8	539.1
3	565.3	541.5
4	564.6	545.6
5	565.0	543.2
6	561.9	535.8
7	565.7	552.0
8	561.8	536.1
9	563.0	537.8
10	564.8	545.6

Estimación del promedio con valores plausibles (vp)

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

```
## [1] 542.4
```

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

```
## [1] 2.823
```

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

Media_vp	Media_vp_se
564.5	23.31

Estimación de la varianza con valores plausibles (vp)

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

Estimación de la varianza con valores plausibles (vp)

vp	promedio	var
1	161566	475622454
2	158646	466558771
3	160375	473836379
4	158913	468536756
5	157276	466068705
6	163288	481896783
7	158017	467768163
8	157597	466331288
9	160005	473369210
10	158220	464190893

Estimación de la varianza con valores plausibles (vp)

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

```
## [1] 470417940
```

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

```
## [1] 3666144
```

```
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
560.8	24.48	173740	23830

```
resultado
```

Media_vp	Media_vp_se	var_vp	var_vp_se
564.5	23.31	159390	21782

Estimación de la proporción con valores plausibles (vp)

```
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)

vp	Employment	prop	prop_var
1	Unemployed	0.0391	0e+00
1	Inactive	0.4073	1e-04
1	Employed	0.5536	2e-04
2	Unemployed	0.0391	0e+00
2	Inactive	0.3986	1e-04
2	Employed	0.5622	2e-04
3	Unemployed	0.0391	0e+00
3	Inactive	0.3875	1e-04
3	Employed	0.5733	2e-04
4	Unemployed	0.0391	0e+00
4	Inactive	0.4147	1e-04
4	Employed	0.5462	2e-04

Estimación de la varianza con valores plausibles (vp)

```
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	0.0491	1e-04
Inactive	0.3972	2e-04
Employed	0.5537	2e-04

```
resultado
```

Employment	prop_pv	Ubar	B	prop_pv_var
Unemployed	0.0391	0e+00	0e+00	0e+00
Inactive	0.4041	1e-04	1e-04	2e-04
Employed	0.5567	2e-04	1e-04	3e-04

Crear valores perdidos ingresos y desocupación (desocupado o ocupado)

Realizar parte practica ## Tabla resumen 12.1 Tablas de resumen antes y despues de imputar. Incluir algunos métodos imputación (no multiple). - na.omit ok - imputación por media ok - imputación por media condicional ok

- hot-deck: Encuesta actual: escoger un donante aleatorio para realizar la imputación. ok
- Cold-deck : Require otras fuentes ok
- Imputación por regresión:
- Vecino más cercano: vector de distancia (gasto)
- Vecino más cercano con regresión. paso 1 regresión paso 2 predicción paso 3 comparar las predicciones de perdidos y no perdidos paso 4 el valor imputado será el valor más próximo a mi valor perdido (términos de predicción) *NOTA* Se toma es el valor observado.

Imputación multiple

Incluir el gráfico 12.5

¿qué es un valor plausible?

Realizar la estimación para ingresos y desocupados.

Mostrar las imputaciónes en la base 10 valores plausibles

incluir sección 12.4.4.1

Incluir las ecuación 12.1 y 12.2 NOTA Hacer el paso a paso ## incluir sección box 12.2

Imputación fraccional.