```
---
title: "Imputación de datos usando el paquete MICE"
author: "Grupo 10"
format: html
---
```

GRUPO 10 - Integrantes

- ANNALISA CATERINA GUTIERREZ UTRILLA

- CARLOS RAMIRO HUARCAYA ANTEZANA

- EDWIN ADRIAN PECEROS ARENAS

- MILUSKA SARAI ZAMBRANO MOTTA

- YEZIT KATERIN QUISPE MONROY

Instalar y cargar los paquetes

```{r}
install.packages("mice")
install.packages("ggmice")
```

```{r}
library(mice)
library(tidyverse)
library(here)
library(rio)
library(ggmice)
library(gtsummary)
```

1 Datos perdidos en investigación en salud

Es común encontrar datos faltantes en un conjunto de datos. Por ejemplo, al recolectar información a partir de historias clínicas de pacientes en un hospital, algunas variables pueden no estar disponibles porque no fueron medidas, anotadas o solicitadas por el personal de salud. En otro escenario, en estudios que utilizan encuestas, es posible que las personas encuestadas no respondan ciertas preguntas o que las respuestas sean ininteligibles.

Cuando se aplican métodos de regresión en investigaciones en ciencias de la salud, la práctica habitual consiste en eliminar las observaciones que contienen datos faltantes. Esta técnica se conoce como análisis de casos completos, y muchos paquetes estadísticos la implementan por defecto.

2 Imputación de datos

Siempre es preferible utilizar todas las observaciones en un análisis de regresión, ya que esto permite obtener estimaciones más precisas y cercanas a la realidad. En esta sesión, aplicaremos una técnica llamada imputación, que consiste en reemplazar los datos perdidos con una estimación de su valor verdadero.

Esta no es una técnica reciente. Enfoques anteriores de imputación —como, por ejemplo, reemplazar los valores perdidos con el promedio de la variable— han sido ampliamente utilizados, pero presentan limitaciones. Estas limitaciones han sido superadas por una técnica más moderna y actualmente muy popular: la imputación múltiple de datos.

3 El dataset para este ejercicio

Para ilustrar el proceso de imputación múltiple de datos, utilizaremos el conjunto de datos data_sm. Este dataset contiene información de 383 pacientes diagnosticados con cáncer diferenciado de tiroides, Las variables registradas comprenden variables demográficas, clínicas, patológicas, de TNM y de seguimiento. Los datos incluyen variables numéricas (como la edad y el tamaño del tumor) y categóricas (como sexo, estado funcional tiroideo, tipo histológico, nivel de riesgo, estado de ganglios linfáticos y presencia de metástasis a distancia) , entre otras. Algunos participantes presentan valores faltantes en al menos una de estas variables.

Cargando los datos

```{r}
data_Tiroides <- import(here("data", "tiroid.csv"))
```

Un vistazo a los datos

```{r}
head(data_Tiroides)
```

Description: df [6 × 17]

	Edad <int>	Genero <chr>	Fumador <chr>	Historia_Fu... <chr>	Historia_Rad... <chr>
1	27	Femenino	No	No	No
2	*NA*	Femenino	*NA*	*NA*	No
3	*NA*	Femenino	*NA*	No	No
4	*NA*	Femenino	No	No	No
5	*NA*	Femenino	No	No	No
6	52	Masculino	Sí	No	No

6 rows | 1-6 of 17 columns

4 Realizando la imputación de datos

4.1 ¿Donde estan los valores perdidos?

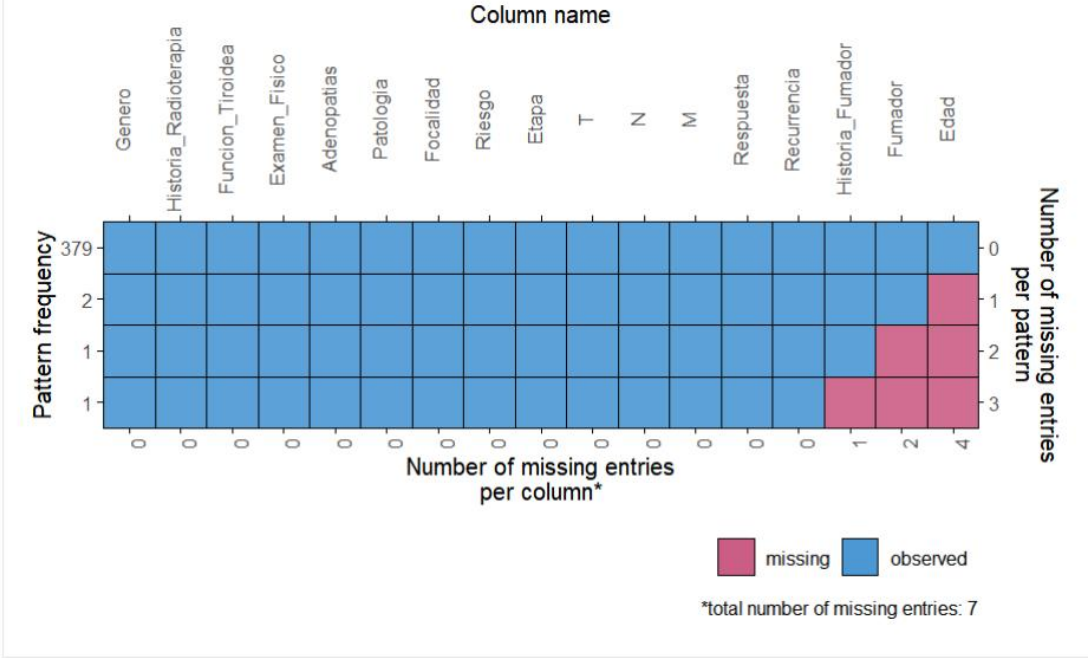
Es importante saber en qué variables se encuentran los datos antes de iniciar la inputación. Una forma rápida es usando la función `colSums()` es `is.na()`.

```{r}
colSums(is.na(data_Tiroides))
```

```
                  Edad                 Genero                Fumador
Historia_Fumador Historia_Radioterapia
                     4                      0                      2
1                    0
     Funcion_Tiroidea         Examen_Fisico           Adenopatias
Patologia              Focalidad
                     0                      0                      0
0                    0
               Riesgo                      T                      N
M                 Etapa
                     0                      0                      0
0                    0
            Respuesta           Recurrencia
                     0                      0
```

Incluso mejor, podemos visualizar los datos perdidos en un mapa de calor usando la función `plot_pattern()` de **ggmice**.

```r
{r}
data_Tiroides |>
  select(
      Edad,
      Genero,
      Fumador,
      Historia_Fumador,
      Historia_Radioterapia,
      Funcion_Tiroidea,
      Examen_Fisico,
      Adenopatias,
      Patologia,
      Focalidad,
      Riesgo,
      Etapa,,
      T,
      N,
      M,
      Respuesta,
      Recurrencia
    ) |>
  ggmice::plot_pattern(
    square = TRUE,
    rotate = TRUE
  )
```



El número total de valores perdidos en el dataset data_Tiroides es de 7. Las variables `Historia_Fumador`, `Fumador` y `Edad` tienen 1, 2 y 4 valores perdidos, respectivamente. Hay un paciente que tiene valores perdidos en dos variables y otro paciente que tiene valores perdidos en 3 variables.

4.2 Comparación de participantes con y sin valores perdidos

Una buena práctica antes de iniciar la imputación de datos es también evaluar cómo difieren los valores de las otras variables entre el grupo de participantes con valores perdidos y el grupo sin valores perdidos. Esto es importante debido a que puede darnos pistas de si en realidad es necesaria la imputación o, dicho de otra forma, si es seguro usar el análisis de casos completos. ¿Cómo? si la distribución de las otras variables no difiere entre el grupo con valores perdidos y el grupo sin valores perdidos, entonces no es necesario la imputación de datos. Evaluemos esto en nuestro dataset para la variable `Fumador` y `Edad`

```{r}
tabla_Fumador = data_Tiroides |>
  dplyr::select(
      Edad,
      Genero,
      Fumador,
      Historia_Fumador,
      Historia_Radioterapia,
      Funcion_Tiroidea,
      Examen_Fisico,
      Adenopatias,
      Patologia,
      Focalidad,
      Riesgo,
      Etapa,,
      T,
      N,
      M,
      Respuesta,
      Recurrencia
    ) |>
  mutate(missing = factor(
    is.na(Fumador),
    levels = c(FALSE, TRUE),
    labels = c("Sin valores perdidos", "Con valores perdidos")
  )) |>
  tbl_summary(
    by = missing,
    statistic = list(
      all_continuous()  ~ "{mean} ({sd})",
      all_categorical() ~ "{n}    ({p}%)")
    ) |>
  modify_header(label = "**Variable**",
                all_stat_cols() ~ "**{level}**<br>N = {n} ({style_percent(p, digits
=1)}%)") |>
  modify_caption("Características de los participantes segun valor perdido") |>
  bold_labels()
```

```r
tabla_Edad = data_Tiroides |>
  dplyr::select(
      Edad,
      Genero,
      Fumador,
      Historia_Fumador,
      Historia_Radioterapia,
      Funcion_Tiroidea,
      Examen_Fisico,
      Adenopatias,
      Patologia,
      Focalidad,
      Riesgo,
      Etapa,,
      T,
      N,
      M,
      Respuesta,
      Recurrencia
    ) |>
  mutate(missing = factor(
    is.na(Edad),
    levels = c(FALSE, TRUE),
    labels = c("Sin valores perdidos", "Con valores perdidos")
  )) |>
  tbl_summary(
    by = missing,
    statistic = list(
      all_continuous()  ~ "{mean} ({sd})",
      all_categorical() ~ "{n}    ({p}%)")
    ) |>
  modify_header(label = "**Variable**",
                all_stat_cols() ~ "**{level}**<br>N = {n} ({style_percent(p, digits
=1)}%)") |>
  modify_caption("Características de los participantes segun valor perdido") |>
  bold_labels()

tabla <- tbl_merge(
  tbls = list(tabla_Fumador, tabla_Edad),
  tab_spanner = c("**Fumador**", "**Edad**")
)
```

```r
tabla
```

Características de los participantes segun valor perdido

Variable	Fumador		Edad	
	Sin valores perdidos N = 381 (99.5%)[1]	**Con valores perdidos** N = 2 (0.52%)[1]	**Sin valores perdidos** N = 379 (99.0%)[1]	**Con valores perdidos** N = 4 (1.04%)[1]
Edad	41 (15)	NA (NA)	41 (15)	NA (NA)
Unknown	2	2	0	4
Genero				
Femenino	310 (81%)	2 (100%)	308 (81%)	4 (100%)
Masculino	71 (19%)	0 (0%)	71 (19%)	0 (0%)
Fumador				

Nota que el promedio y desviación estandard, para algunas variables, varían en la comparación del grupo con variables perdidas y completas.

4.3 ¿Qué variables debo incluir en el proceso de imputación?

Debemos incluir todas las variables que se utilizarán en los análisis posteriores, incluso aquellas que no presentan valores perdidos. La razón es que el modelo de imputación debe ser *tan complejo como el análisis que se realizará posteriormente*. De lo contrario, se perderá información relevante de las demás variables. Además, aunque algunas variables no tengan valores faltantes, su inclusión en el modelo de imputación es útil porque aportan información que mejora la estimación de los valores imputados. Recuerda además que las variables categóricas deben ser de tipo factor. El código de abajo selecciona las variables y transforma la variable `Historia_Fumador` a factor.

```{r}
input_data =
  data_Tiroides |>
    dplyr::select(
      Edad,
      Genero,
      Fumador,
      Historia_Fumador,
      Historia_Radioterapia,
      Funcion_Tiroidea,
      Examen_Fisico,
      Adenopatias,
      Patologia,
      Focalidad,
      Riesgo,
      Etapa,,
      T,
      N,
      M,
      Respuesta,
      Recurrencia
    ) |>
  mutate(Historia_Fumador = as.factor(Historia_Fumador))
```

4.4 La función `mice()` para imputar datos

Para imputar datos utilizaremos la función `mice()` del paquete del mismo nombre. Entre sus argumentos, debemos especificar:

- el número de imputaciones con `m`,
- una semilla (`seed`) para que los resultados sean reproducibles, y
- el método de imputación con `method`.

Con respecto a este último argumento, emplearemos el método `"pmm"` para variables continuas y `"logreg"` para variables binarias. Para las variables que **no presentan valores perdidos**, simplemente se colocan comillas vacías (`""`).

Cabe recalcar que el conjunto de datos contiene 17 variables, de las cuales 3 presentan valores perdidos, y las variables se encuentran en el siguiente orden.

```{r}
names(input_data)
```

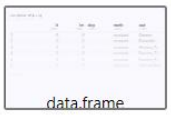
```
 [1] "Edad"                "Genero"              "Fumador"
"Historia_Fumador"     "Historia_Radioterapia"
 [6] "Funcion_Tiroidea"    "Examen_Fisico"       "Adenopatias"
"Patologia"            "Focalidad"
[11] "Riesgo"              "Etapa"               "T"                   "N"
"M"
[16] "Respuesta"           "Recurrencia"
```

El método de imputación la indicaremos con el argumento `method` en el mismo orden que aparecen las variables en el dataset.

```{r}
data_imputada =
  mice(
    input_data,
    m = 20,
    method = c(
      "pmm",
      "",
      "logreg",
      "logreg",
      "",
      "",
      "",
      "",
      "",
      "",
      "",
      "",
      "",
      "",
      "",
      "",
      "",
      "",
      "",
      ""),
    maxit = 20,
    seed = 3,
    print = F
    )
```

```
Aviso: Number of logged events: 15
```

```{r}
data_imputada
```



Description: df [6 × 5]

	it <dbl>	im <dbl>	dep <chr>	meth <chr>	out <chr>
1	0	0		constant	Genero
2	0	0		constant	Fumador
3	0	0		constant	Historia_Radio...
4	0	0		constant	Funcion_Tiroid...
5	0	0		constant	Examen_Fisico
6	0	0		constant	Adenopatias

6 rows

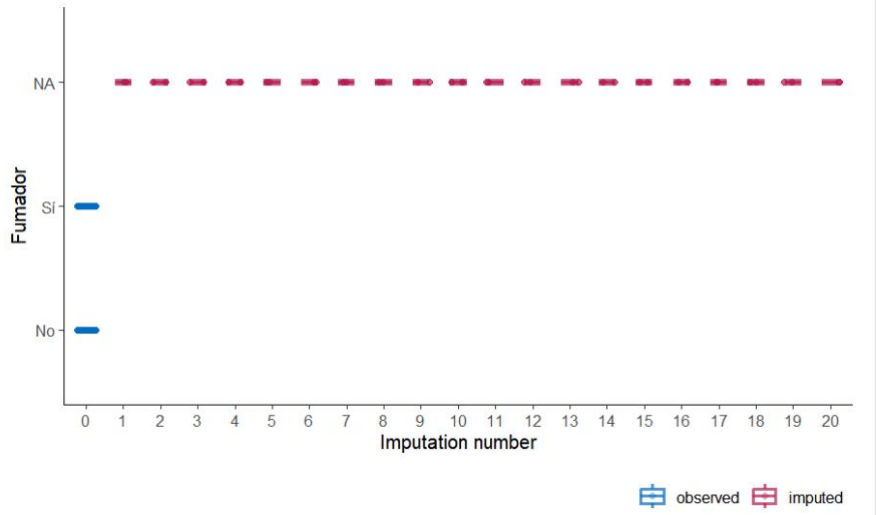
El resultado de la imputación se ha guardado en el objeto data_imputada y muestra que es un objeto de clase mids (multiply imputed dataset), el número de imputaciones (20), el método de imputación para todas las variables, y en una matriz, cuales variables han sido usadas para predecir otras.

5 Analizando los datos imputados

Antes de realizar análisis adicionales al dataset imputado, es necesario explorar los datos imputados. Idealmente, los valores imputados deben ser plausibles en comparación con los valores observados. Podemos observar esto en un gráfico de cajas y bigotes de la distribución de los datos imputados (20) versus los datos sin imputar.

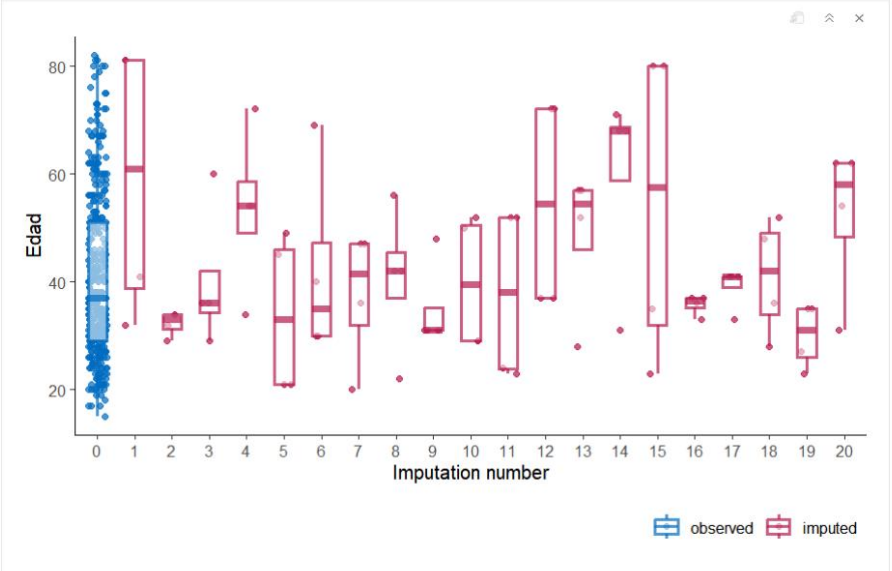
Para la variable Fumador

```{r}
ggmice(data_imputada, aes(x = .imp, y = Fumador)) +
  geom_jitter(height = 0, width = 0.25) +
  geom_boxplot(width = 0.5, size = 1, alpha = 0.55, outlier.shape = NA) +
  labs(x = "Imputation number")
```



Para la variables Edad

```{r}
ggmice(data_imputada, aes(x = .imp, y = Edad)) +
  geom_jitter(height = 0, width = 0.25) +
  geom_boxplot(width = 0.5, size = 1, alpha = 0.55, outlier.shape = NA) +
  labs(x = "Imputation number")
```



Con esta función, los datos observados se encuentran al inicio (azul), y los demás boxplots corresponden a los datos imputados (20). Para ambos casos, los datos imputados estan dentro del rango de los valores observados, son plausibles.

Para datos categóricos, podemos crear una tabla de dos entradas comparando la distribución de la variable con datos completos e incompletos. Esto requiere primero crear la versión "long" de la data imputada.

```{r}
data_imputada_l <- complete(data_imputada, "long", include = TRUE)
```

Ahora la tabla.

```{r}
data_imputada_l <- data_imputada_l %>%
  mutate(imputed = .imp > 0,
         imputed = factor(imputed,
                          levels = c(8044,8044),
                          labels = c("Observado", "Imputado")))

prop.table(table(data_imputada_l$Historia_Fumador,
                 data_imputada_l$imputed),
           margin = 2)
```

```
    Observado Imputado
No
Sí
```

Idealmente los dos primero número luego del decimal, debe ser similares entre datos observados e imputados.

5.1 Procedimientos adicionales luego de la imputación

El procedimiento estándar para realizar un análisis de regresión después de la imputación consiste en utilizar la función `with()` para ajustar el modelo de regresión al objeto `mids` (por ejemplo, `data_imputada`). Posteriormente, se emplea la función `pool()` para obtener los resultados combinados, como se suele presentar en la sección de resultados.

No obstante, si se hace uso del paquete **gtsummary**, este y sus funciones manejan internamente el agrupamiento de las imputaciones, por lo que solo es necesario utilizar la función `with()`. A continuación, se muestra un ejemplo de regresión logística multivariada con los datos imputados, tal como lo realizaste anteriormente.

Recuerda que es posible realizar cualquier tipo de análisis de regresión o (con procedimientos adicionales) pruebas inferenciales a partir de los datos imputados.

```{r}
tabla_multi <-
  data_imputada |>
  with(glm(Historia_Fumador ~ Edad + Genero + Fumador +
             Historia_Radioterapia + Funcion_Tiroidea + Examen_Fisico + Adenopatias
+ Patologia + Focalidad + Riesgo + Etapa + T + N + M + Respuesta + Recurrencia,
           family = binomial(link = "logit"))) |>
  tbl_regression(exponentiate = TRUE,
                 label = list(
                   Edad ~ "Edad del paciente en años al momento del diagnóstico",
                   Genero ~ "Sexo del paciente",
                   Fumador ~ "Consumo actual de tabaco",
                   Historia_Radioterapia ~ "Exposición previa a radioterapia en
región cervical", Funcion_Tiroidea ~ "Estado funcional de la glándula tiroides",
                   Examen_Fisico ~ "Hallazgos en la exploración física tiroidea",
                   Adenopatias ~ "Presencia y localización de adenopatías
cervicales", Patologia ~ "Tipo histológico del carcinoma",
                   Focalidad ~ "Número de focos tumorales",
                   Riesgo ~ "Estratificación de riesgo",
                   Etapa ~ "Estadificación por sistem AJCC (I, II, III, IVA, IVB)",
             T ~ "Extensión del tumor primario (T1a, T1b, T2, T3a, T3b, T4a, T4b)",
                   N ~ "Afectación ganglionar regional (N0, N1a, N1b)",
                   M ~ "Presencia de metástasis a distancia (M0, M1)",
                   Respuesta ~ "Respuesta al tratamiento inicial",
                   Recurrencia ~ "Recurrencia de la enfermedad")) |>
  bold_p(t = 0.05) |>
  modify_header(estimate = "**OR ajustado**", p.value = "**p valor** ")
```

```{r}
tabla_multi
```

Characteristic	OR ajustado	95% CI	p valor
Edad del paciente en años al momento del diagnóstico	0.96	0.90, 1.02	0.2
Sexo del paciente			
Femenino	—	—	
Masculino	1.26	0.15, 10.7	0.8
Consumo actual de tabaco			
No	—	—	
Sí	10.9	0.96, 124	0.054
Exposición previa a radioterapia en región cervical			

```
---
title: "Imputación de datos usando el paquete MICE"
author: "Grupo 10"
format: html
---
```

GRUPO 10 - Integrantes

- ANNALISA CATERINA GUTIERREZ UTRILLA

- CARLOS RAMIRO HUARCAYA ANTEZANA

- EDWIN ADRIAN PECEROS ARENAS

- MILUSKA SARAI ZAMBRANO MOTTA

- YEZIT KATERIN QUISPE MONROY

Instalar y cargar los paquetes

```
{r}
install.packages("mice")
install.packages("ggmice")
```

```
{r}
library(mice)
library(tidyverse)
library(here)
```

GRUPO 10 - Integrantes
Instalar y cargar los …
1 Datos perdidos en i…
2 Imputación de datos
3 El dataset para est…
4 Realizando la impu…
4.1 ¿Donde estan l…
4.2 Comparación d…
4.3 ¿Qué variables …
4.4 La función mice…
5 Analizando los dat…
5.1 Procedimientos …

```
R 4.4.3 · C:/Users/MVaya/Downloads/
+   modify_header(estimate = "**OR ajustado**", p.value = "**p valor** ")
Aviso: glm.fit: fitted probabilities numerically 0 or 1 occurredAviso: glm.fit: fitted probabilities numerically 0 or 1 occurredAvis
o: glm.fit: fitted probabilities numerically 0 or 1 occurredAviso: glm.fit: fitted probabilities numerically 0 or 1 occurredAviso: g
lm.fit: fitted probabilities numerically 0 or 1 occurredAviso: glm.fit: fitted probabilities numerically 0 or 1 occurredAviso: glm.f
it: fitted probabilities numerically 0 or 1 occurredAviso: glm.fit: fitted probabilities numerically 0 or 1 occurredAviso: glm.fit:
fitted probabilities numerically 0 or 1 occurredAviso: glm.fit: fitted probabilities numerically 0 or 1 occurredAviso: glm.fit: fitt
ed probabilities numerically 0 or 1 occurredAviso: glm.fit: fitted probabilities numerically 0 or 1 occurredAviso: glm.fit: fitted p
robabilities numerically 0 or 1 occurredAviso: glm.fit: fitted probabilities numerically 0 or 1 occurredAviso: glm.fit: fitted proba
bilities numerically 0 or 1 occurredAviso: glm.fit: fitted probabilities numerically 0 or 1 occurredAviso: glm.fit: fitted probabili
ties numerically 0 or 1 occurredAviso: glm.fit: fitted probabilities numerically 0 or 1 occurredAviso: glm.fit: fitted probabilities
numerically 0 or 1 occurredAviso: glm.fit: fitted probabilities numerically 0 or 1 occurred
> tabla_multi
> View(tabla_multi)
> 
```

Data

data_imputada	List of 23
data_imputada…	8043 obs. of 20 variables
data_Tiroides	383 obs. of 17 variables
input_data	383 obs. of 17 variables
tabla	Large tbl_merge (4 elements, 21.7 MB)
tabla_Edad	Large tbl_summary (5 elements, 10.8 …
tabla_Fumador	Large tbl_summary (5 elements, 10.8 …
tabla_multi	Large tbl_regression (6 elements, 13…

D: > UPSJB_Practica_RStudio > estadistica_upsjb > data

Name	Size	Modified
..		
Visualizar.Datos.Grupo1.qmd	13.7 KB	May 3, 2025, 9:34 PM
nombre_de_objeto.png	52.8 KB	May 3, 2025, 9:34 PM
Grupo10.PC4.Actividad.1.pdf	3 MB	Jul 2, 2025, 2:43 PM
grupo 10 pc4 act1 doc.docx	3 MB	Jul 2, 2025, 2:43 PM
almac_sangr.csv	30.4 KB	Jun 28, 2025, 8:24 AM
almac_sangre.csv	38.3 KB	Apr 17, 2025, 9:37 PM
diabetes.csv	28.5 KB	Apr 17, 2025, 9:37 PM
tiroides(1).csv	51.3 KB	Jul 2, 2025, 1:19 PM
tiroides.csv	49.9 KB	Jul 2, 2025, 12:13 PM
.Rhistory	5.3 KB	May 3, 2025, 2:20 PM
.RData	140.7 KB	May 3, 2025, 2:18 PM