```
---
title: "Imputación de datos usando el paquete MICE"
author: "Grupo 01"
format: html
---
```

GRUPO 01 - Integrantes

- Bastidas Bendezu Ivan Luis
- Basurto Taype Marcelo Aaron
- Fonseca Moron José Kenneth
- Saravia Gutierrez Randy Esteban
- Talavera Ayllon Angel Ronaldo

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)
```

```
Aviso: package 'mice' was built under R version 4.5.1
```

```
Adjuntando el paquete: 'mice'

The following object is masked from 'package:stats':

    filter

The following objects are masked from 'package:base':

    cbind, rbind
```

```
── Attaching core tidyverse packages ──────────────────────── tidyverse 2.0.0 ──
✓ dplyr     1.1.4     ✓ readr     2.1.5
✓ forcats   1.0.0     ✓ stringr   1.5.1
✓ ggplot2   3.5.2     ✓ tibble    3.2.1
✓ lubridate 1.9.4     ✓ tidyr     1.3.1
✓ purrr     1.0.4
── Conflicts ────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks mice::filter(), stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package to force all conflicts to become errors
```

```
here() starts at D:/Programas
Universidad/UPSJB_Practica_RStudio/estadistica_upsjb
Some optional R packages were not installed and therefore some file formats are
not supported. Check file support with show_unsupported_formats()
```

```
Aviso: package 'ggmice' was built under R version 4.5.1
```

```
Adjuntando el paquete: 'ggmice'

The following objects are masked from 'package:mice':

    bwplot, densityplot, stripplot, xyplot
```

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 `almac_sangre`. Este dataset incluye información de 316 pacientes adultos. Las variables registradas comprenden la edad mediana de globulos rojos, la edad, la raza afroamericana (si o no), la historia familiar, el estadio T, el volumen prostatico y el volumen tumoral, entre otras. Algunos participantes presentan valores faltantes en al menos una de estas variables.

Cargando los datos

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

```
Registered S3 method overwritten by 'data.table':
  method           from
  print.data.table
```

Un vistazo a los datos

```{r}
head(almac_sangre)
```

Description: df [6 × 20]

	Grupo_edad_GR <chr>	Edad_mediana... <int>	Edad <dbl>	Raza_afroame... <chr>
1	Mayor	25	72.1	No
2	Mayor	25	73.6	No
3	Mayor	25	67.5	No
4	Intermedio	15	65.8	No
5	Intermedio	15	63.2	No
6	Mayor	25	65.4	No

6 rows | 1-5 of 20 columns

4 Realizando la imputación de datos

4.1 ¿Donde estan los valores perdidos?

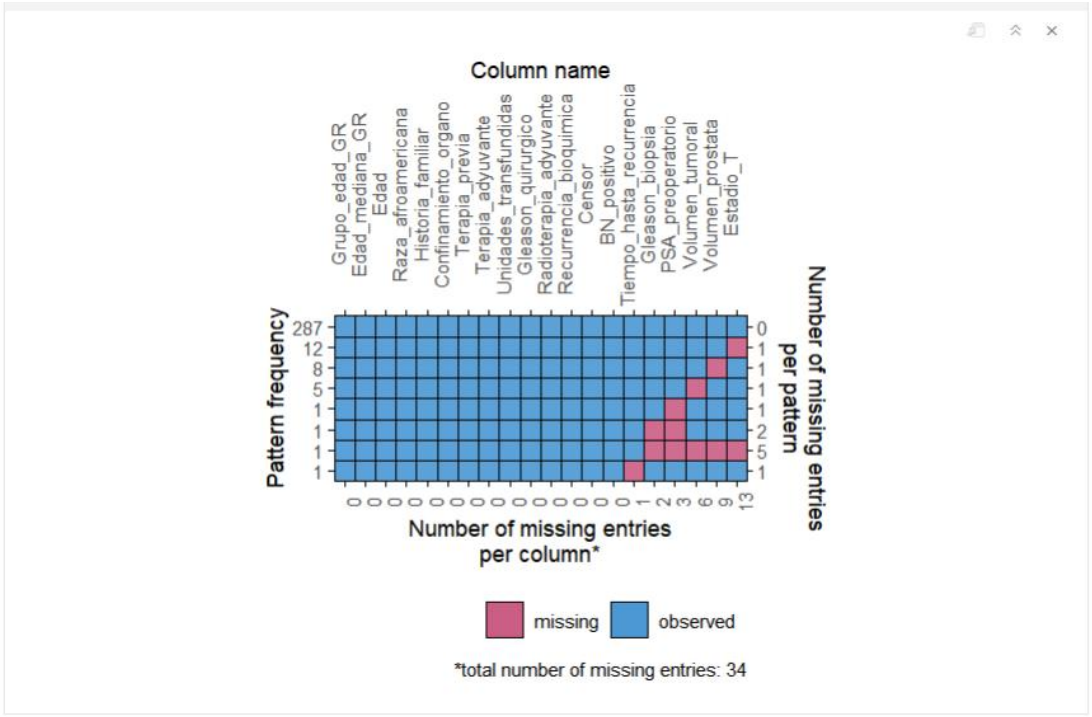
Es importante saber en qué variables se encuentran los datos antes de iniciar la inputación. Una forma rápida es usando la función `colSums()` es `is.na()`.

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

```
           Grupo_edad_GR          Edad_mediana_GR                     Edad
Raza_afroamericana
                       0                        0                        0
0
       Historia_familiar         Volumen_prostata          Volumen_tumoral
Estadio_T
                       0                        9                        6
13
         Gleason_biopsia     Confinamiento_organo        PSA_preoperatorio
Terapia_previa
                       2                        0                        3
0
  Unidades_transfundidas       Gleason_quirurgico        Terapia_adyuvante
Radioterapia_adyuvante
                       0                        0                        0
0
  Recurrencia_bioquimica                   Censor Tiempo_hasta_recurrencia
BN_positivo
                       0                        0                        1
0
```

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

```{r}
almac_sangre |>
  select(
      Grupo_edad_GR,
      Edad_mediana_GR,
      Edad,
      Raza_afroamericana,
      Historia_familiar,
      Volumen_prostata,
      Volumen_tumoral,
      Estadio_T,
      Gleason_biopsia,
      Confinamiento_organo,
      PSA_preoperatorio,
      Terapia_previa,
      Terapia_adyuvante,
      Unidades_transfundidas,
      Gleason_quirurgico,
      Radioterapia_adyuvante,
      Recurrencia_bioquimica,
      Censor,
      Tiempo_hasta_recurrencia,
      BN_positivo
    ) |>
  ggmice::plot_pattern(
    square = TRUE,
    rotate = TRUE
  )
```



El número total de valores perdidos en el dataset almac_sangre es de 34. Las variables `Tiempo_hasta_recurrencia`, `Gleason_biopsia` y `PSA_preoperatoria` , `Volumen_tumoral` , `Volumen_prostata` , `Estadio_T` tienen 1, 2, 3, 6, 9 y 13 valores perdidos, respectivamente. Hay 1 paciente que tiene valores perdidos en dos variables y otro paciente que tiene valores perdidos en 5 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 `Gleason_biopsisa` e `PSA_preoperatorio`

```{r}
tabla_Gleason_biopsia = almac_sangre |>
  dplyr::select(
      Grupo_edad_GR,
      Edad_mediana_GR,
      Edad,
      Raza_afroamericana,
      Historia_familiar,
      Volumen_prostata,
      Volumen_tumoral,
      Estadio_T,
      Gleason_biopsia,
      Confinamiento_organo,
      PSA_preoperatorio,
      Terapia_previa,
      Terapia_adyuvante,
      Unidades_transfundidas,
      Gleason_quirurgico,
      Radioterapia_adyuvante,
      Recurrencia_bioquimica,
      Censor,
      Tiempo_hasta_recurrencia,
      BN_positivo
    ) |>
  mutate(missing = factor(
    is.na(Gleason_biopsia),
    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_PSA_preoperatorio = almac_sangre |>
  dplyr::select(
      Grupo_edad_GR,
      Edad_mediana_GR,
      Edad,
      Raza_afroamericana,
      Historia_familiar,
      Volumen_prostata,
      Volumen_tumoral,
      Estadio_T,
      Gleason_biopsia,
      Confinamiento_organo,
      PSA_preoperatorio,
      Terapia_previa,
      Terapia_adyuvante,
      Unidades_transfundidas,
      Gleason_quirurgico,
```

```
      Radioterapia_adyuvante,
      Recurrencia_bioquimica,
      Censor,
      Tiempo_hasta_recurrencia,
      BN_positivo
    ) |>
  mutate(missing = factor(
    is.na(PSA_preoperatorio),
    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_Gleason_biopsia, tabla_PSA_preoperatorio),
  tab_spanner = c("**Gleason_biopsia**", "**PSA_preoperatorio**")
)
```

```
{r}
tabla
```

Características de los participantes segun valor perdido

Variable	Gleason_biopsia		PSA_preoperatorio	
	Sin valores perdidos N = 314 (99.4%)[1]	Con valores perdidos N = 2 (0.63%)[1]	Sin valores perdidos N = 313 (99.1%)[1]	Con valores perdidos N = 3 (0.95%)[1]
Grupo_edad_GR				
Intermedio	101 (32%)	2 (100%)	101 (32%)	2 (67%)
Joven	106 (34%)	0 (0%)	106 (34%)	0 (0%)
Mayor	107 (34%)	0 (0%)	106 (34%)	1 (33%)
Edad_mediana_GR				
10	106 (34%)	0 (0%)	106 (34%)	0 (0%)

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 `Estadio_T` a factor.

```{r}
input_data =
  almac_sangre |>
    dplyr::select(
      Grupo_edad_GR,
      Edad_mediana_GR,
      Edad,
      Raza_afroamericana,
      Historia_familiar,
      Volumen_prostata,
      Volumen_tumoral,
      Estadio_T,
      Gleason_biopsia,
      Confinamiento_organo,
      PSA_preoperatorio,
      Terapia_previa,
      Terapia_adyuvante,
      Unidades_transfundidas,
      Gleason_quirurgico,
      Radioterapia_adyuvante,
      Recurrencia_bioquimica,
      Censor,
      Tiempo_hasta_recurrencia,
      BN_positivo
    ) |>
  mutate(Estadio_T = as.factor(Estadio_T))
```

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 20 variables, de las cuales 6 presentan valores perdidos, y las variables se encuentran en el siguiente orden.

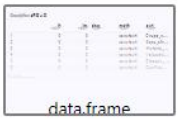
```{r}
names(input_data)
```

```
 [1] "Grupo_edad_GR"             "Edad_mediana_GR"           "Edad"
"Raza_afroamericana"
 [5] "Historia_familiar"         "Volumen_prostata"          "Volumen_tumoral"
"Estadio_T"
 [9] "Gleason_biopsia"           "Confinamiento_organo"      "PSA_preoperatorio"
"Terapia_previa"
[13] "Terapia_adyuvante"         "Unidades_transfundidas"    "Gleason_quirurgico"
"Radioterapia_adyuvante"
[17] "Recurrencia_bioquimica"    "Censor"
"Tiempo_hasta_recurrencia" "BN_positivo"
```

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",
      "pmm",
      "logreg",
      "pmm",
      "",
      "pmm",
      "",
      "",
      "",
      "",
      "",
      "",
      "",
      "",
      "pmm",
      ""),
    maxit = 20,
    seed = 3,
    print = F
    )
```

```{r}
data_imputada
```

Description: df [6 × 5]

	it <dbl>	im <dbl>	dep <chr>	meth <chr>	out <chr>
1	0	0		constant	Grupo_edad...
2	0	0		constant	Raza_afroa...
3	0	0		constant	Historia_fam...
4	0	0		constant	Volumen_tu...
5	0	0		constant	Gleason_bio...
6	0	0		constant	Confinamien...

6 rows

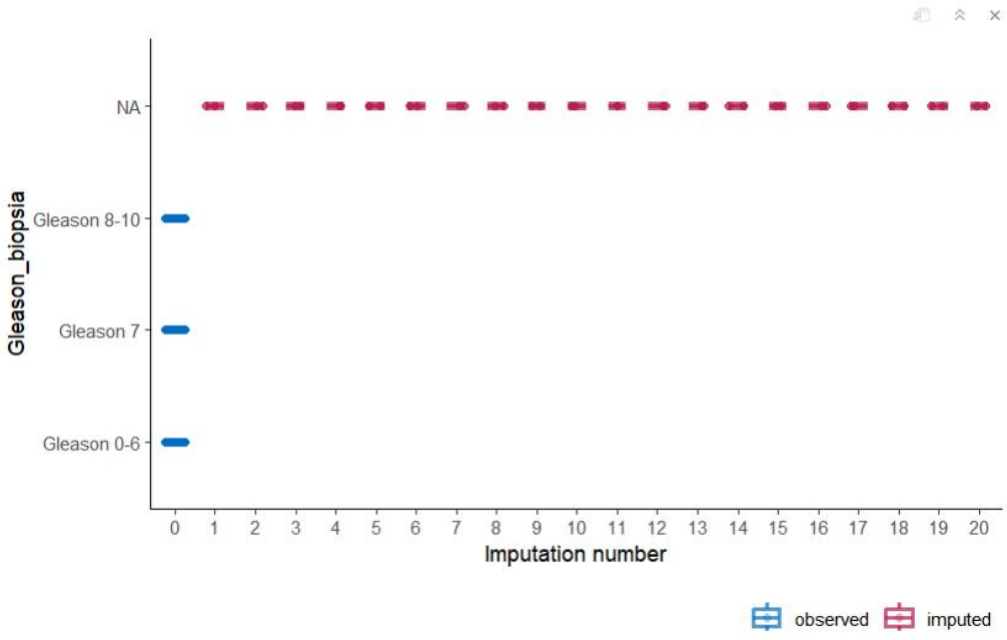
El resultado de la imputación se ha guardado en el objeto data_imputada y muestra que es un objeto de clase mids (multiply imputed dataset), el número de imputaciones (20), el método de imputación para todas las variables, y en una matriz, cuales variables han sido usadas para predecir otras.

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.

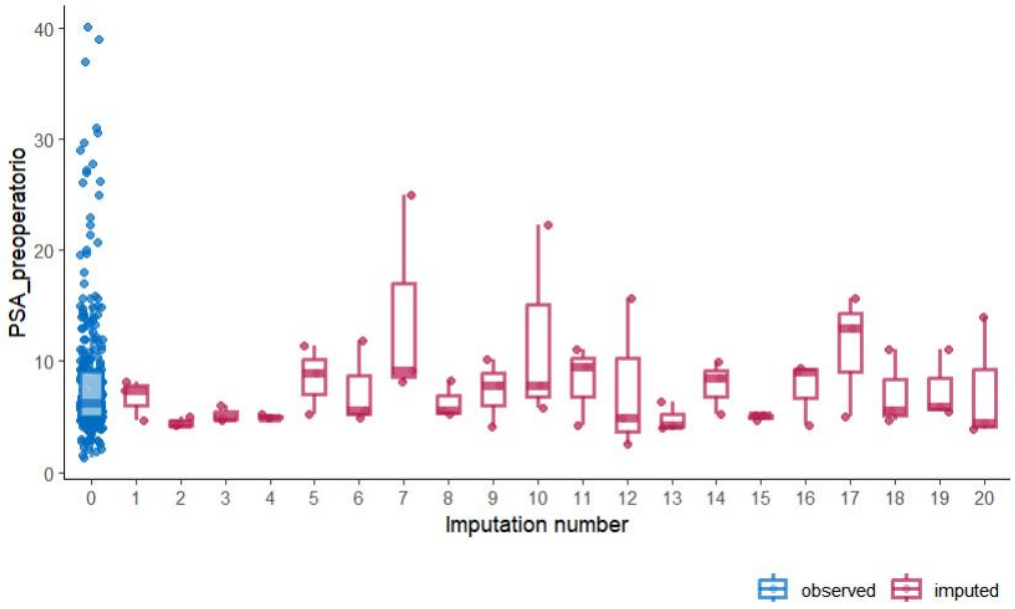
Para la variable Gleason_biopsia

```{r}
ggmice(data_imputada, aes(x = .imp, y = Gleason_biopsia)) +
  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 PSA_preoperatorio

```{r}
ggmice(data_imputada, aes(x = .imp, y = PSA_preoperatorio)) +
  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(F,T),
                          labels = c("Observado", "Imputado")))

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

```
        Observado  Imputado
T1-T2a 0.8877888 0.8865506
T2b-T3 0.1122112 0.1134494
```

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

5.1 Procedimientos adicionales luego de la imputación

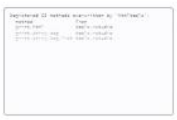
El procedimiento estándar para realizar un análisis de regresión después de la imputación consiste en utilizar la función `with()` para ajustar el modelo de regresión al objeto `mids` (por ejemplo, `data_imputada`). Posteriormente, se emplea la función `pool()` para obtener los resultados combinados, 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(Estadio_T ~ Grupo_edad_GR + Edad_mediana_GR + Edad +
             Raza_afroamericana + Historia_familiar + Volumen_prostata +
Volumen_tumoral + Gleason_biopsia + Confinamiento_organo + PSA_preoperatorio +
Terapia_previa + Terapia_adyuvante + Unidades_transfundidas + Gleason_quirurgico +
Radioterapia_adyuvante + Recurrencia_bioquimica + Censor + Tiempo_hasta_recurrencia
+ BN_positivo,
           family = binomial(link = "logit"))) |>
  tbl_regression(exponentiate = TRUE,
                 label = list(
                   Grupo_edad_GR ~ "Grupo de duración de almacenamiento de GR",
                   Edad_mediana_GR ~ "Edad mediana de GR transfundidas",
                   Edad ~ "Edad del paciente",
                   Raza_afroamericana ~ "Indica si es de raza afroamericana",
                   Historia_familiar ~ "Historia familiar de la enfermedad",
                   Volumen_prostata ~ "Volumen de la próstata",
                   Volumen_tumoral ~ "Volumen del tumor",
                   Gleason_biopsia ~ "Puntuación Gleason de la biopsia",
                   Confinamiento_organo ~ "Tumor confinado al órgano",
                   PSA_preoperatorio ~ "Antígeno prostático específico
preoperatorio",
                   Terapia_previa ~ "Terapia preoperatoria recibida",
                   Terapia_adyuvante ~ "Recibió terapia adyuvante",
                   Unidades_transfundidas ~ "Número de unidades alogénicas
transfundidas",
                   Gleason_quirurgico ~ "Puntuación Gleason quirúrgica",
                   Radioterapia_adyuvante ~ "Recibió radioterapia adyuvante",
                   Recurrencia_bioquimica ~ "Recurrencia bioquímica del cáncer de
próstata",
                   Censor ~ "Indicador de censura",
                   Tiempo_hasta_recurrencia ~ "Tiempo hasta la recurrencia
bioquímica",
                   BN_positivo ~ "Cuello vesical positivo")) |>
  bold_p(t = 0.05) |>
  modify_header(estimate = "**OR ajustado**", p.value = "**p valor** ")
```

```{r}
tabla_multi
```



Characteristic	OR ajustado	95% CI	p valor
Grupo de duración de almacenamiento de GR			
Intermedio	—	—	
Joven	4.52	0.99, 20.6	0.051
Mayor	2.40	0.49, 11.7	0.3
Edad mediana de GR transfundidas			
Edad del paciente	1.07	0.99, 1.16	0.085
Indica si es de raza afroamericana			
No	—	—	
Sí	0.22	0.04, 1.25	0.086