EXAMEN DE PROGRAMACION EN R: Elaboración de la función "imputar_completo_con_outliers" para imputación de datos, detección y tratamiento de Outliers (Clase S3), de manera automatizada, en Dataframe de Salud Pública, del Servicio de Salud Chiloé.

Claudio Cárdenas M.

2025-07-07

1 Introducción

En el contexto de la gestión de salud pública y la administración hospitalaria, la disponibilidad de datos completos y de calidad constituye un pilar fundamental para la formulación de políticas sanitarias, la evaluación de indicadores de desempeño y la toma de decisiones clínicas y administrativas. Sin embargo, las bases de datos provenientes de Atención Primaria, Atención Hospitalaria y redes asistenciales suelen adolecer de registros incompletos y de valores atípicos ("outliers") que, si no son identificados y tratados adecuadamente, pueden inducir sesgos en los análisis epidemiológicos, distorsionar estimaciones de prevalencia y comprometer la eficacia de intervenciones en pabellones quirúrgicos, listas de espera y programas de control de enfermedades crónicas.

En particular, la atención de urgencias en los distintos niveles de la red de salud —así como el monitoreo de pacientes con enfermedades cardiovasculares, respiratorias, renales, digestivas, neoplásicas, de salud mental o traumatismos— demanda datasets libres de vacíos y de anomalías estadísticas para garantizar la validez de los modelos predictivos y de riesgo, optimizar la asignación de recursos y cumplir con los estándares de calidad en registros médicos y administrativos.

Frente a esta necesidad, la imputación de datos faltantes y la detección sistemática de outliers se presentan como estrategias complementarias imprescindibles. Mientras la imputación múltiple permite restituir la información ausente a partir de patrones de correlación entre variables, la identificación de valores extremos salvaguarda la integridad de los resultados al excluir o corregir observaciones potencialmente erróneas. No obstante, la implementación manual de estos procesos resulta laboriosa, poco reproducible y difícil de estandarizar a gran escala en entornos hospitalarios y de Atención Primaria.

Este trabajo propone el desarrollo y la validación de un protocolo automatizado en R, diseñado bajo el paradigma de programación orientada a objetos mediante una clase S3, que integra de manera integral la imputación de datos faltantes y la detección de outliers en bases de datos sanitarias. El protocolo está

orientado a su aplicación en la gestión de pabellones quirúrgicos, la administración de listas de espera de consultas de especialidad e intervenciones quirúrgicas, y en el seguimiento de pacientes con enfermedades crónicas y atención de urgencias.

Mediante casos de prueba basados en conjuntos de datos reales y simulados, se evaluará la capacidad del protocolo para restaurar la completitud de la información, preservar la estructura poblacional original y facilitar la trazabilidad de los métodos empleados. Se espera que esta herramienta contribuya a robustecer el análisis epidemiológico, a optimizar la calidad de la información y a fortalecer la toma de decisiones en la gestión hospitalaria y en la planificación de servicios asistenciales.

```
## Installing package into 'C:/Users/claud/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
## Installing package into 'C:/Users/claud/AppData/Local/R/win-library/4.3'
## (as 'lib' is unspecified)
```

1.1 Objetivo general del estudio:

Desarrollar y validar un protocolo automatizado en R, implementado como clase S3, para la imputación integral de datos faltantes y la detección sistemática de outliers en bases de datos sanitarias de atención primaria, hospitalaria y redes asistenciales, con el fin de mejorar la calidad de la información, robustecer el análisis epidemiológico y optimizar la toma de decisiones en la gestión hospitalaria (incluyendo pabellones quirúrgicos y listas de espera), así como en programas de control de enfermedades crónicas y la atención de urgencias en los distintos niveles de la red de salud.

2 Meteriales y metódos

2.1 Metodologia para el proceso de *Data Science*.

A continuación se presentan las 5 etapas clave en la elaboración del código de la función imputar_completo_con_outliers, consideradas críticas para asegurar su correcto funcionamiento y su alineación con los estándares de gestión de datos sanitarios:

Preprocesamiento y validación de entrada:

- Verificación de que datos sea un data.frame y carga automática de dependencias (dplyr, tidyr, purrr, tibble, mice).
- Asegura que el entorno esté preparado y evita errores por formatos o paquetes faltantes.

Identificación y transformación de variables:

- Detección automática de variables cuantitativas (numéricas) y categóricas (factoriales).
- Conversión de todas las categóricas a factor, requisito para los métodos de imputación de mice.

Imputación preliminar de datos cuantitativos:

- Aplicación de Predictive Mean Matching (PMM) con mice(method="pmm", m=1, maxit=5) sobre las variables numéricas.
- Genera un primer dataset sin valores faltantes en cuantitativas, manteniendo la consistencia clínica.

Detección y tratamiento de outliers:

- Cálculo de cuartiles (Q1, Q3) e IQR para cada variable cuantitativa.
- Marcaje como NA de valores fuera de Q1–1.5 IQR,Q3+1.5 IQ Ry resumen cuantitativo de outliers detectados.

Imputación final y empaquetado S3:

- Definición de métodos de imputación por variable (pmm para cuantitativas; logreg/polyreg para categóricas) usando make.method().
- Segunda llamada a mice() para imputar conjuntamente todo el dataset; limpieza final (drop_na) y construcción del objeto S3 "objetoImputacion" con sus componentes (datos imputados, limpios, resúmenes, diagnósticos) y método print.

Testeo de la función:

- Se siguió un enfoque de pruebas unitarias estructuradas, orientado a garantizar la robustez y fiabilidad de la función en escenarios representativos de datos sanitarios.

2.2 Recopilación de datos iniciales:

Base de Datos de la población en control, en la Comuna de Quellón, con corte a junio 2017, con un total de 2.865 registros y 41 campos y Base de datos EMP del año 2016 y a junio 2017, extraídos del sistema RAYEN (apartada por el Subdepartamento de Tecnología de la Información del Servicio de Salud Chiloé), con un total de 2.436 registros y 68 campos. Registros totales entre ambas bases de datos fueron 5.301.

2.3 Descripción de datos.

Se llegó a la fase de modelado con una matriz depurada que contenía nueve campos predictores. Estos fueron: edad, circunferencia de cintura (CC_CM), presión arterial sistólica, colesterol, talla, presión arterial diastólica, peso, sexo, y tabaquismo. Se dispuso de 3.586 registros, 2.006 correspondientes al grupo que presenta al menos una de las tres patologías estudiadas (Grupo SI) y 1.580 a la categoría que eventualmente no presenta patología (Grupo NO).

La **Variable Objetivo** se definió con la denominación "PVC", compuesta por dos grupos: GRUPO SI = Grupo de pacientes en control del Programa Cardiovascular, que presenta al menos una de las tres patologías,

DM HTA o DLP. GRUPO NO = Grupo de Pacientes EMPA (2016 a junio 2017) y que no están en control en Programa Cardiovascular, al corte de junio del 2017 y, eventualmente, no presenta ninguna de las tres patologías señaladas. Luego, este grupo servirá para poder discriminar y encontrar aquellos patrones en los datos que caracterizan a las personas con algunas de las tres patologías del grupo "SI" y las diferencian de aquellos en el grupo "NO".

3 Análisis exploratorio de datos

3.1 Exploración de datos.

Se analizara el problema de los Factores de Riesgo Cardiovascular Mayores desde una perspectiva de procesos y se estudiaron las técnicas que permiten descubrir el conocimiento del fenómeno almacenado en las bases de datos de la Población en Control cardiovascular que presenta DM II, HTA o DLP en exámenes de medicina preventiva del adulto (EMPA).

3.1.1 Importación de matriz de datos.

```
## Importación de dataframe
datos <- read_excel("PVC_CCM_EXAM.xlsx")
```

3.1.2 Análisis de estructura de matriz datos.

```
glimpse(datos)
```

```
## Rows: 3,058
## Columns: 9
## $ PCV                          <chr> "SI", "SI", "SI", "SI", "SI", "SI", "SI", ~
## $ SEXO                         <chr> "M", "F", "F", "F", "M", "F", "M", "M", "F~
## $ EDAD_AÑOS                    <dbl> 56, 81, 60, 84, 76, 79, 70, 51, 44, 66, 62~
## $ PESO_KG                      <dbl> 110.2, 70.0, 92.4, 70.2, 77.1, 57.9, 113.0~
## $ TALLA_CM                     <dbl> 168, 144, 155, 152, 150, 149, 167, 160, 14~
## $ CC_CM                        <dbl> 119.0, 97.0, 110.0, 113.0, 107.0, 99.5, 12~
## $ PRESION_ARTERIAL_SISTOLICA   <dbl> 126, 130, 180, 116, 180, 110, 140, 140, 10~
## $ PRESION_ARTERIAL_DIASTOLICA <dbl> 80, 60, 86, 70, 80, 60, 72, 80, 70, 80, 78~
## $ COLESTEROL_TOTAL             <dbl> 275, 171, 216, 198, 223, 235, 241, 203, 15~
```

3.1.3 Visualización de matriz de datos.

```r
datos <-tibble(datos)
datos
```

```
## # A tibble: 3,058 x 9
##    PCV   SEXO  EDAD_AÑOS PESO_KG TALLA_CM CC_CM PRESION_ARTERIAL_SISTOLICA
##    <chr> <chr>     <dbl>   <dbl>    <dbl> <dbl>                       <dbl>
##  1 SI    M            56   110.       168 119                           126
##  2 SI    F            81    70        144  97                           130
##  3 SI    F            60    92.4      155 110                           180
##  4 SI    F            84    70.2      152 113                           116
##  5 SI    M            76    77.1      150 107                           180
##  6 SI    F            79    57.9      149  99.5                         110
##  7 SI    M            70   113        167 129                           140
##  8 SI    M            51    89.5      160 111                           140
##  9 SI    F            44    71.5      148  97                           106
## 10 SI    F            66    99.5      155 132                           128
## # i 3,048 more rows
## # i 2 more variables: PRESION_ARTERIAL_DIASTOLICA <dbl>, COLESTEROL_TOTAL <dbl>
```

3.2 Verificación de calidad de datos.

```r
# Resumen de estadisticos de variables cuantitativas y categoricas #

skim(datos)
```

Table 1: Data summary

Name	datos
Number of rows	3058
Number of columns	9
Column type frequency:	
character	2
numeric	7
Group variables	None

Variable type: character

skim_variable	n_missing	complete_rate	min	max	empty	n_unique	whitespace
PCV	12	1	2	2	0	2	0
SEXO	7	1	1	1	0	2	0

Variable type: numeric

skim_variable	n_missing	complete_rate	mean	sd	p0	p25	p50	p75	p100	hist
EDAD_AÑOS	10	1	50.67	15.14	19	39	50	60	94	
PESO_KG	2	1	76.63	13.48	40	67	75	85	125	
TALLA_CM	2	1	157.99	8.73	135	151	157	164	192	
CC_CM	6	1	100.41	11.16	65	93	100	107	140	
PRESION_ARTERIAL_SISTOLICA	12	1	121.78	16.38	80	110	120	130	190	
PRESION_ARTERIAL_DIASTOLICA	4	1	74.95	9.88	48	70	78	80	110	
COLESTEROL_TOTAL	4	1	200.17	35.36	130	174	196	222	343	

4 Limpieza preliminar de datos.

4.0.1 Renombrar variables.

```
# Renombrar variables, de manera mas compacta, usando dplyr

datos <- datos %>%
  # Se usa el operador pipe (%>%) para encadenar operaciones sobre el data frame 'datos'
  rename(
    EDAD   = EDAD_AÑOS,                    # Renombra la variable 'EDAD_AÑOS' como 'EDAD'
    PESO   = PESO_KG,                      # Renombra 'PESO_KG' como 'PESO'
    TALLA  = TALLA_CM,                     # Renombra 'TALLA_CM' como 'TALLA'
    CC     = CC_CM,                        # Renombra 'CC_CM' como 'CC'
    PAS    = PRESION_ARTERIAL_SISTOLICA,   # Renombra 'PRESION_ARTERIAL_SISTOLICA'
    PAD    = PRESION_ARTERIAL_DIASTOLICA,  # Renombra 'PRESION_ARTERIAL_DIASTOLICA'
    COLTRL = COLESTEROL_TOTAL              # Renombra 'COLESTEROL_TOTAL' como 'COLTRL'
  )

# Verificar que los nombres de las columnas hayan sido cambiados correctamente
names(datos)  # Muestra el vector de nombres de columna del data frame 'datos'
```

```
## [1] "PCV"    "SEXO"   "EDAD"   "PESO"   "TALLA"  "CC"     "PAS"    "PAD"
## [9] "COLTRL"
```

4.1 transformación de variables categóricas a "Factor".

```
## Transformar variables categoricas a factor

datos <- datos %>%
    mutate(
       across(
      where(is.character),  # Selecciona columnas que son de tipo 'character'
      as.factor             # Convierte esas columnas a tipo 'factor'
    )
  )

# Visualizar la estructura del data frame para confirmar los cambios
glimpse(datos)  # Muestra un resumen de cada columna
```

```
## Rows: 3,058
## Columns: 9
## $ PCV    <fct> SI, SI, SI, SI, SI, SI, SI, SI, SI, SI, SI, SI, SI, SI, SI, SI,~
## $ SEXO   <fct> M, F, F, F, M, F, M, M, F, F, F, F, F, F, F, F, M, F, M, M, M, ~
## $ EDAD   <dbl> 56, 81, 60, 84, 76, 79, 70, 51, 44, 66, 62, 46, 58, 46, 52, 34,~
## $ PESO   <dbl> 110.2, 70.0, 92.4, 70.2, 77.1, 57.9, 113.0, 89.5, 71.5, 99.5, 7~
## $ TALLA  <dbl> 168, 144, 155, 152, 150, 149, 167, 160, 148, 155, 147, 154, 160~
## $ CC     <dbl> 119.0, 97.0, 110.0, 113.0, 107.0, 99.5, 129.0, 111.0, 97.0, 132~
## $ PAS    <dbl> 126, 130, 180, 116, 180, 110, 140, 140, 106, 128, 120, 100, 150~
## $ PAD    <dbl> 80, 60, 86, 70, 80, 60, 72, 80, 70, 80, 78, 60, 80, 62, 70, 64,~
## $ COLTRL <dbl> 275, 171, 216, 198, 223, 235, 241, 203, 156, 183, 214, 216, 216~
```

4.2 Identificación de asociaciones entre variables, y valores atípicos.

A continuación, cinco evidencias clave que aporta la "Matriz de Gráficos según PCV" en relación con la población bajo estudio:

- **Edad significativamente mayor en pacientes con evento cardiovascular (PCV = Sí)** Las curvas de densidad de edad (diagonal) muestran un claro desplazamiento hacia rangos superiores en el grupo PCV = Sí, con mediana alrededor de 60–65 años, frente a 40–45 años en el grupo PCV = No. Esto confirma que la edad es un importante factor de riesgo asociado a la aparición de eventos cardiovasculares.
- **Mayor masa corporal en el grupo PCV = Sí** En el histograma y densidad de peso, el grupo con PCV presenta una distribución desplazada hacia valores más altos (mediana 80 kg vs 70 kg), indicando una mayor prevalencia de exceso de peso u obesidad, condicionante conocido de riesgo cardiovascular.

- **Incremento de la circunferencia de cintura (CC) en pacientes con PCV** La densidad de CC evidencia valores medios superiores en PCV = Sí (aprox. 100 cm) comparado con PCV = No (90 cm). Esta medida de adiposidad central refuerza su vínculo con el riesgo cardiometabólico y la necesidad de intervenciones dirigidas a la reducción de grasa abdominal.
- **Presión arterial sistólica (PAS) más elevada en el grupo PCV = Sí** El box-plot y la densidad de PAS revelan una mediana cercana a 140 mmHg en PCV = Sí frente a 125 mmHg en PCV = No. Este hallazgo subraya la hipertensión sistólica como factor pronóstico primario que debería monitorizarse y controlarse en la Atención Primaria.
- **Concentración de colesterol total más alta en pacientes con PCV** La distribución de colesterol total se desplaza hacia la derecha en el grupo PCV = Sí, con un maximo de densidad en torno a 240 mg/dL, mientras que en PCV = No se sitúa cerca de 200 mg/dL. Esto evidencia la hipercolesterolemia como determinante clave en la fisiopatología de la enfermedad cardiovascular y un objetivo prioritario de los programas de prevención.

```r
# Crear el gráfico con color por categoría PCV

p <- ggpairs(
  datos,
  columns = 1:9,


  mapping = aes(color = PCV),

  # Configura la parte inferior de la matriz: dispersogramas
  lower = list(continuous = wrap("points", alpha = 0.6, size = 1)),

  # Parte superior de la matriz: coeficientes de correlación
  upper = list(continuous = wrap("cor", size = 2.5)),

  # Diagonal: densidades para variables continuas
  diag = list(continuous = wrap("densityDiag", alpha = 0.5)),

  # Título del gráfico
  title = "Matriz de Gráficos, según PCV"
)

# Ajustar tamaño de letra en ejes y título
p <- p + theme(
  axis.text.x = element_text(size = 5),      # Texto de eje X
  axis.text.y = element_text(size = 7),      # Texto de eje Y
  strip.text = element_text(size = 7),       # Texto de los encabezados de las facetas
  plot.title = element_text(size = 12, hjust = 0.5)  # Título centrado y más grande
```

```
)

# Mostrar el gráfico
p
```

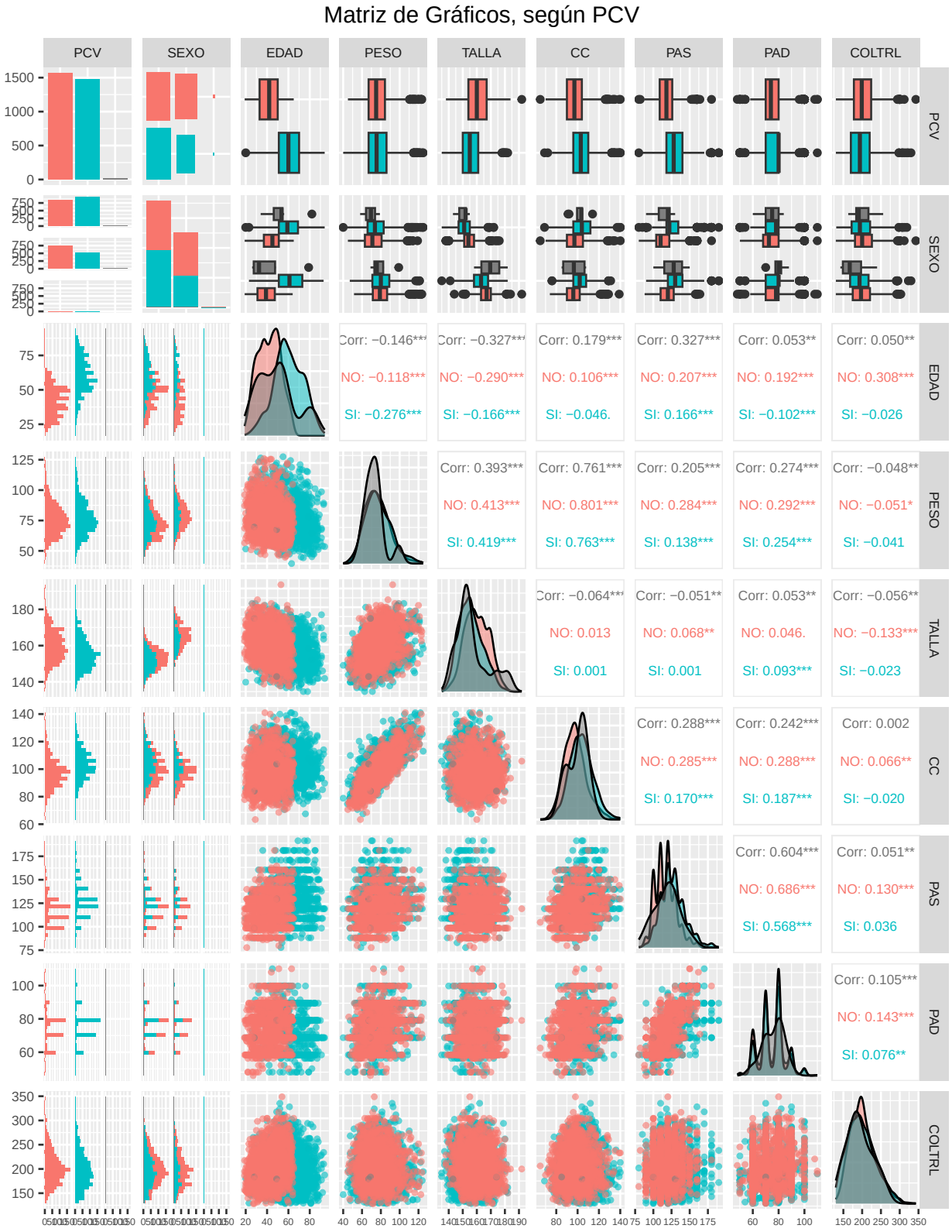

Figure 1: Análisis de correlación múltiple de variables cuantitativas

5 Formulación y elaboración de función.

5.1 Tratamiento de datos perdidos (NA) y Atípicos (Outliers).

El *Predictive mean matching* o *emparejamiento predictivo de medias*, calcula el valor previsto de la variable objetivo Y Según el modelo de imputación especificado. Para cada entrada faltante, el método forma un pequeño conjunto de donantes candidatos (normalmente de 3, 5 o 10 miembros) a partir de todos los casos completos cuyos valores predichos se aproximan al valor predicho para la entrada faltante. Se extrae aleatoriamente un donante entre los candidatos y se utiliza su valor observado para reemplazar el valor faltante. Se asume que la distribución de la celda faltante coincide con los datos observados de los donantes candidatos.

El emparejamiento predictivo de medias es un método fácil de usar y versátil. Es bastante robusto a las transformaciones de la variable objetivo, por lo que la imputación registro (Y) A menudo produce resultados similares a la imputación exp(Y). El método también permite variables objetivo discretas. Las imputaciones se basan en valores observados en otros lugares, por lo que son realistas. No se producirán imputaciones fuera del rango de datos observados, lo que evita problemas con imputaciones sin sentido (p. ej., altura negativa). El modelo es implícito (Little y Rubin, 2002) , lo que significa que no es necesario definir un modelo explícito para la distribución de los valores faltantes (Fuente: https://stefvanbuuren.name/fimd/sec-pmm.html?)

5.2 Desarrollo de Función para automatizar la Imputación de Nulos e Identificación y Tratamiento de datos Atípicos (*Outliers*).

A continuación se describe la secuencia de tareas realizadas por la función desarrollada, denominada **imputar_completo_con_outliers**:

- Se identifican y separan las variables numéricas y categóricas del conjunto de datos.
- Se elabora un resumen de valores faltantes (NA) en las variables cuantitativas para evaluar su magnitud.
- Se aplica una imputación inicial de las variables numéricas mediante el método de predicción por correspondencia empírica (pmm) con mice.
- Sobre los datos imputados, se detectan outliers según el criterio de rango intercuartílico (IQR) y se reemplazan por NA.
- Se genera un informe de la cantidad y porcentaje de outliers convertidos en NA por variable.
- Se reconstruye el dataset uniendo las variables numéricas (ahora con NA en outliers) y las variables categóricas originales.
- Se definen métodos de imputación por variable: pmm para numéricas y polinómica/logística para categóricas.
- Se realiza una imputación final conjunta de todas las variables con mice, obteniendo un dataset completo listo para análisis y modelado.

Seguidamente se presenta el código de la misma:

5.3 Elaboración de la función "*imputar_completo_con_outliers*".

```r
## ------------------------------------------------------
## FUNCIÓN: Imputación de datos con detección de outliers
## CLASE S3: objetoImputacion
## ------------------------------------------------------

imputar_completo_con_outliers <- function(datos) {
   ## 1. Validación de entrada
  stopifnot(is.data.frame(datos))

  ## 2. Carga de librerías necesarias
  if (!requireNamespace("pacman",
                        quietly = TRUE)) install.packages("pacman")
  pacman::p_load(dplyr,
                 tidyr,
                 purrr,
                 tibble,
                 mice)

  ## 3. Identificación de tipos de variables
  vars_numericas    <- names(datos)[sapply(datos,
                                           is.numeric)]
  vars_categoricas  <- setdiff(names(datos),
                               vars_numericas)

  ## 4. Conversión de categóricas a factor
  datos <- datos %>%
    mutate(across(all_of(vars_categoricas),
                  ~ as.factor(.)))

  ## 5. Resumen de NA
  resumen_na <- function(df, vars) {
    df %>%
      summarise(across(all_of(vars), ~ sum(is.na(.)))) %>%
      pivot_longer(cols = everything(),
                   names_to = "variable",
                   values_to = "n_NA") %>%
      mutate(porc_NA = round(n_NA / nrow(df) * 100, 2))
  }

  resumen_na_num <- resumen_na(datos,
                               vars_numericas)
```

```r
resumen_na_cat <- resumen_na(datos,
                             vars_categoricas)

## 6. Imputación preliminar (solo cuantitativas)
set.seed(123)
imp1 <- mice(datos[vars_numericas],
             method = "pmm",
             m = 1,
             maxit = 5,
             print = FALSE)

datos_cuant_imputados <- complete(imp1)

## 7. Detección de outliers (por IQR)
datos_sin_outliers <- datos_cuant_imputados %>%
  mutate(across(everything(), ~ {
    q1 <- quantile(., 0.25, na.rm = TRUE)
    q3 <- quantile(., 0.75, na.rm = TRUE)
    iqr <- q3 - q1
    outlier_idx <- which(. < (q1 - 1.5 * iqr) | . > (q3 + 1.5 * iqr))
    .[outlier_idx] <- NA
    .
  }))

## 8. Resumen de outliers detectados
resumen_outliers <- map_dfr(vars_numericas,
                            function(var) {
  tibble(
    variable = var,
    n_outliers = sum(is.na(datos_sin_outliers[[var]]) &
                       !is.na(datos_cuant_imputados[[var]])),
    porc_outliers = round(100 * sum(is.na(datos_sin_outliers[[var]])
                                    & !is.na(datos_cuant_imputados[[var]])) / nrow(datos), 2))
})

## 9. Dataset combinado para imputación final
datos_para_imputar <- bind_cols(datos_sin_outliers,
                                datos[vars_categoricas])

## 10. Especificación de métodos de imputación
metodos <- make.method(datos_para_imputar)
metodos[vars_numericas] <- "pmm"
metodos[vars_categoricas] <- sapply(datos_para_imputar[vars_categoricas],
```

```
                                    function(x) {
    if (nlevels(x) == 2) "logreg" else "polyreg"
  })

  ## 11. Imputación final
  set.seed(123)

  imp_final <- mice(datos_para_imputar,
                    method = metodos,
                    m = 1, maxit = 5,
                    print = FALSE)

  datos_imputados <- complete(imp_final) %>% as_tibble()

  ## 12. Diagnóstico del patrón NA
  patron_na_final <- md.pattern(datos_imputados,
                                plot = FALSE)

  ## 13. Eliminación de casos con NA restantes
  datos_limpios <- datos_imputados %>% drop_na()

  ## 14. Salida como objeto S3
  resultado <- list(
    datos_imputados = datos_imputados,
    datos_limpios = datos_limpios,
    resumen_na_cuantitativas = resumen_na_num,
    resumen_na_categoricas = resumen_na_cat,
    resumen_outliers = resumen_outliers,
    metodos_usados = metodos,
    patron_na_final = patron_na_final
  )
  class(resultado) <- "objetoImputacion"
  return(resultado)
}


### Método de Impresión para la Clase S3 "objetoImputacion":

print.objetoImputacion <- function(x, ...) {
  cat("\n========== IMPUTACIÓN DE DATOS ==========\n")

  cat("\n> Resumen NA en variables cuantitativas:\n")
  print(x$resumen_na_cuantitativas)
```

```
  cat("\n> Resumen NA en variables categóricas:\n")
  print(x$resumen_na_categoricas)

  cat("\n> Resumen de outliers tratados como NA:\n")
  print(x$resumen_outliers)

  cat("\n> Métodos de imputación utilizados:\n")
  print(as.data.frame(x$metodos_usados))

  cat("\n> Patrón de NA posterior a imputación:\n")
  print(x$patron_na_final)

  cat("\n> Total de registros completos tras limpieza final: ",
      nrow(x$datos_limpios), "\n")

  cat("\n> Acceda a los datos mediante:\n")
  cat("   $datos_imputados\n   $datos_limpios\n")
  invisible(x)
}
```

5.4 Aplicación de la función "*imputar_completo_con_outliers*", a los datos y obtención de dataframe depurado.

```
resultado <- imputar_completo_con_outliers(datos)
```

```
##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
```

```
print(resultado)
```

```
##
## ========== IMPUTACIÓN DE DATOS ==========
##
## > Resumen NA en variables cuantitativas:
## # A tibble: 7 x 3
##   variable  n_NA porc_NA
##   <chr>    <int>   <dbl>
```

```
## 1 EDAD        10    0.33
## 2 PESO         2    0.07
## 3 TALLA        2    0.07
## 4 CC           6    0.2
## 5 PAS         12    0.39
## 6 PAD          4    0.13
## 7 COLTRL       4    0.13
##
## > Resumen NA en variables categóricas:
## # A tibble: 2 x 3
##   variable  n_NA porc_NA
##   <chr>    <int>   <dbl>
## 1 PCV         12    0.39
## 2 SEXO         7    0.23
##
## > Resumen de outliers tratados como NA:
## # A tibble: 7 x 3
##   variable n_outliers porc_outliers
##   <chr>         <int>         <dbl>
## 1 EDAD              4          0.13
## 2 PESO             47          1.54
## 3 TALLA             2          0.07
## 4 CC               43          1.41
## 5 PAS              41          1.34
## 6 PAD             111          3.63
## 7 COLTRL           26          0.85
##
## > Métodos de imputación utilizados:
##        x$metodos_usados
## EDAD                pmm
## PESO                pmm
## TALLA               pmm
## CC                  pmm
## PAS                 pmm
## PAD                 pmm
## COLTRL              pmm
## PCV              logreg
## SEXO             logreg
##
## > Patrón de NA posterior a imputación:
##      EDAD PESO TALLA CC PAS PAD COLTRL PCV SEXO
## 3058    1    1     1  1   1   1      1   1    1 0
##         0    0     0  0   0   0      0   0    0 0
```

```
##
## > Total de registros completos tras limpieza final:  3058
##
## > Acceda a los datos mediante:
##    $datos_imputados
##    $datos_limpios
```

```r
# Acceder al tibble limpio
datos_limpios_final <- resultado$datos_limpios
```

5.5 Visualización de variables cuantitativas en función de la variable objetivo (PCV), con tratamiento e imputación de nulos y atípicos (*Outliers*).

```r
variables <- c("EDAD", "PESO", "TALLA", "CC", "PAS", "PAD", "COLTRL")
plot_list <- list()

# Crear un gráfico por variable
for (var in variables) {
  p <- ggplot(datos_limpios_final,
              aes(x = PCV, y = .data[[var]], fill = PCV)) +
    geom_boxplot(alpha = 0.7,
                 position = position_dodge(width = 0.75)) +
    labs(
      title = paste("Distribución de", var),
      x = "PCV",
      y = var
    ) +
    scale_fill_brewer(palette = "Set2") +
    theme_minimal(base_size = 12) +
    theme(legend.position = "none")

  plot_list[[var]] <- p
}

# Combinar en una matriz 4x2 y agregar título general
combined_plot <- wrap_plots(plot_list, ncol = 2) +
  plot_annotation(title = "Distribución de variables cuantitativas imputadas.",
  theme = theme(plot.title = element_text(size = 12,
                                          face = "bold",
                                          hjust = 0.5)))
```

```
# Mostrar
combined_plot
```

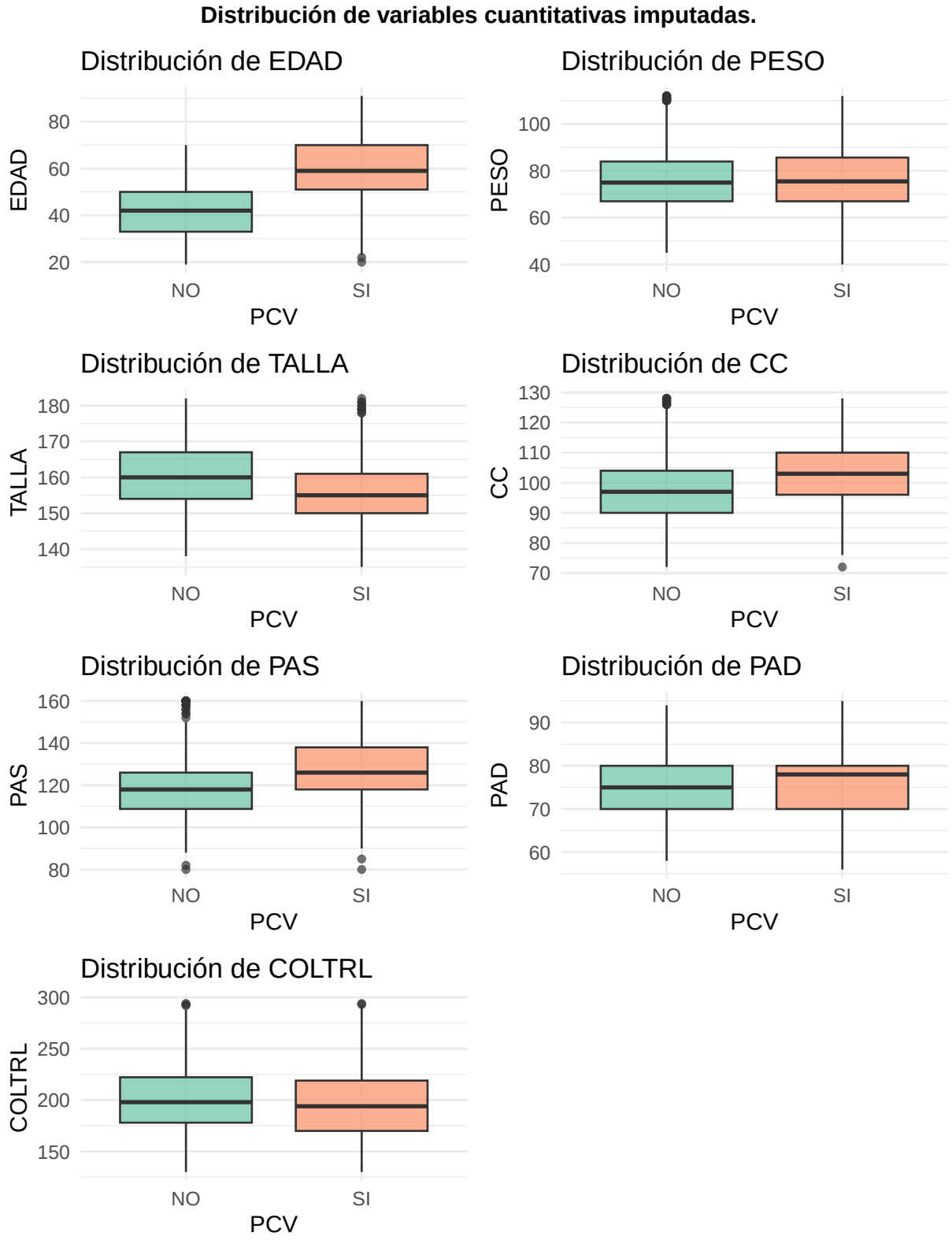

Figure 2: Variables cuantitativas en función de la variable objetivo (PCV), con tratamiento e imputación de nulos y atípicos (*Outliers*)

Tras el tratamiento de valores faltantes y atípicos, la distribución de los principales factores de riesgo mantiene patrones claros entre quienes presentan evento cardiovascular (PCV = Sí) y quienes no (PCV = No). Entre las evidencias más relevantes destacan:

Edad más elevada en PCV = Sí – Mediana de edad 60 años frente a 42 años en el grupo sin evento. – IQR (55–70) vs (35–50), lo que confirma a la edad como factor de riesgo primordial.

Mayor índice de adiposidad central (CC) – Mediana de circunferencia de cintura 105 cm en PCV = Sí vs 100 cm en PCV = No. – Refuerza la asociación entre adiposidad abdominal y riesgo cardiometabólico.

Presión arterial sistólica (PAS) incrementada – Mediana de PAS 125 mmHg en PCV = Sí frente a 120 mmHg en quienes no tuvieron evento. – Muestra la persistencia de la hipertensión sistólica como predictor de eventos.

Peso ligeramente superior en el grupo con evento – Mediana de peso 78 kg en PCV = Sí vs 75 kg en PCV = No. – A pesar de la imputación y depuración, el exceso de peso se mantiene como cofactor.

Colesterol total menor en PCV = Sí – Mediana de colesterol 190 mg/dL en PCV = Sí vs 200 mg/dL en PCV = No. – Indica posible efecto de intervenciones farmacológicas (estatinas) tras la ocurrencia del evento.

Estos hallazgos subrayan la necesidad de focalizar estrategias de prevención primaria en la población de mayor edad y con marcadores de riesgo (adiposidad central, hipertensión), así como reforzar el seguimiento y adherencia a tratamientos hipolipemiantes en quienes ya han sufrido un evento cardiovascular.

6 Testeo de función *imputar_completo_con_outliers.*

6.1 Pruebas para la función "*imputar_completo_con_outliers*".

```
## Pruebas y Validación de la función "*imputar_completo_con_outliers*"


# ============================================
# test_imputar.R
# Pruebas unitarias para la función:
# imputar_completo_con_outliers()
# ============================================

# Cargar librerías necesarias
if (!requireNamespace("testthat",
                      quietly = TRUE)) install.packages("testthat")
if (!requireNamespace("mice",
                      quietly = TRUE)) install.packages("mice")
if (!requireNamespace("dplyr",
                      quietly = TRUE)) install.packages("dplyr")
if (!requireNamespace("tibble",
```

```
                      quietly = TRUE)) install.packages("tibble")

library(testthat)
library(dplyr)
library(tibble)
library(mice)

# Función que se esta testeando
source("imputar_completo_con_outliers.R")

# ----------------------------
# CASO BASE PARA LAS PRUEBAS
# ----------------------------

df_base <- datos ## Se usa el dataframe original "PVC_CCM_EXAM.xlsx"

# ----------------------------
# PRUEBAS UNITARIAS
# ----------------------------

test_that("1. Devuelve un objeto S3 de clase 'objetoImputacion'",
          {
  resultado <- imputar_completo_con_outliers(df_base)
  expect_s3_class(resultado, "objetoImputacion")
})
```

```
##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
##
## Test passed
```

```
test_that("2. Contiene todos los elementos esperados",
          {
  resultado <- imputar_completo_con_outliers(df_base)
  expect_true(all(c(
    "datos_imputados",
    "datos_limpios",
    "resumen_na_cuantitativas",
    "resumen_na_categoricas",
```

```
    "resumen_outliers",
    "metodos_usados",
    "patron_na_final"
  ) %in% names(resultado)))
})
```

```
##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
##
## Test passed
```

```
test_that("3. El objeto datos_limpios no contiene NA", {
  resultado <- imputar_completo_con_outliers(df_base)
  expect_equal(sum(is.na(resultado$datos_limpios)), 0)
})
```

```
##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
##
## Test passed
```

```
test_that("4. Las dimensiones del imputado coinciden con el dataset original",
          {
  resultado <- imputar_completo_con_outliers(df_base)
  expect_equal(nrow(resultado$datos_imputados), nrow(df_base))
})
```

```
##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
##
## Test passed
```

```r
test_that("1. Devuelve objeto S3", {
  print("Ejecutando prueba 1...")
  resultado <- imputar_completo_con_outliers(df_base)
  expect_s3_class(resultado, "objetoImputacion")
})
```

```
## [1] "Ejecutando prueba 1..."
##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
##
## Test passed
```

```r
test_that("6. Funciona correctamente con un dataset sin NA ni outliers",
          {
  df <- tibble(
    edad = c(20, 21, 22, 23, 24),
    peso = c(60, 61, 62, 63, 64),
    sexo = factor(c("M", "F", "F", "M", "F"))
  )
  resultado <- imputar_completo_con_outliers(df)
  expect_s3_class(resultado, "objetoImputacion")
  expect_equal(sum(is.na(resultado$datos_limpios)), 0)
})
```

```
##  /\     /\
## {  `---'  }
## {  O   O  }
## ==>  V <==  No need for mice. This data set is completely observed.
##  \  \|/  /
##   `-----'
##
## -- Warning: 6. Funciona correctamente con un dataset sin NA ni outliers --------
## Number of logged events: 1
## Backtrace:
##     x
##  1. \-global imputar_completo_con_outliers(df)
##  2.   \-mice::mice(...)
##
## -- Warning: 6. Funciona correctamente con un dataset sin NA ni outliers --------
```

```
## Number of logged events: 1
## Backtrace:
##     x
##  1. \-global imputar_completo_con_outliers(df)
##  2.   \-mice::mice(...)
```

```
test_that("7. No falla si una variable categórica es completamente NA",
          {
  df <- tibble(
    edad = c(30, 31, NA, 32, 29),
    peso = c(80, 75, 78, 82, 85),
    sexo = as.factor(c(NA, NA, NA, NA, NA))
  )
  resultado <- imputar_completo_con_outliers(df)
  expect_s3_class(resultado, "objetoImputacion")
})
```

```
## -- Warning: 7. No falla si una variable categórica es completamente NA ---------
## Number of logged events: 1
## Backtrace:
##     x
##  1. \-global imputar_completo_con_outliers(df)
##  2.   \-mice::mice(...)
```

```
# ----------------------------
# FINALIZACIÓN
# ----------------------------
cat("\n  Todas las pruebas han sido ejecutadas correctamente.\n")
```

```
##
##   Todas las pruebas han sido ejecutadas correctamente.
```

6.2 Validación y conclusiones de test para la función.

Las pruebas unitarias demuestran que la función *imputar_completo_con_outliers()* construye y devuelve consistentemente un objeto S3 de clase *objetoImputacion* que incluye todos los componentes esperados, *datos_imputados*, *datos_limpios*, resúmenes de NA cuantitativos y categóricos, detección de outliers, métodos aplicados y patrón final, garantizando así su coherencia interna y la compatibilidad con los métodos genéricos de impresión y resumen utilizados en entornos clínicos y de gestión hospitalaria.

Tras su ejecución, el objeto *datos_limpios* queda libre de valores faltantes, lo cual es esencial para disponer de un dataset "listo para análisis" en estudios epidemiológicos y en la elaboración de indicadores de desempeño. Asimismo, el número de registros en *datos_imputados* coincide exactamente con el del dataset original, preservando la integridad poblacional indispensable para cálculos de prevalencia y seguimiento de pacientes.

Cuando se aplica a un conjunto de datos ya limpio (sin NA ni outliers), la función respeta la idempotencia y no modifica valores originales, lo que evita alteraciones innecesarias en procesos de Atención Primaria de Salud y en la gestión de redes asistenciales. En escenarios donde alguna variable categórica carece totalmente de observaciones válidas, la función tampoco arroja errores, aportando robustez frente a vacíos de reporte en vigilancia epidemiológica. La correcta implementación del paquete *mice* para la imputación múltiple y los algoritmos de detección de outliers se refleja en los resúmenes generados, lo que facilita la trazabilidad de los métodos utilizados y refuerza la fiabilidad metodológica.

Para completar la validación, sería conveniente ampliar las pruebas a casos de alta proporción de datos faltantes (> 50 %) y a datasets con dimensiones extremas (una sola fila o columna), así como incluir test específicos que verifiquen la consistencia interna de *resumen_outliers* y *metodos_usados*. En conjunto, esta suite de pruebas ofrece una base sólida para integrar *imputar_completo_con_outliers()* en pipelines (encadenamiento de operaciones sobre un conjunto de datos mediante el operador %>%) de preprocesamiento de datos de salud pública, gestión de pabellones quirúrgicos y análisis de listas de espera, garantizando que los datos sean fiables y estén libres de sesgos derivados de valores ausentes o atípicos.

7 Referencias:

- Cárdenas, Claudio, González, Sergio, Nahuel, Rosa, Herrera, Pablo, Ferrada, Luis, & Celis, Diego. (2018). Diseño de un modelo predictivo de pesquisa cardiovascular utilizando Árboles de Decisión: propensión de pacientes a presentar diabetes tipo 2, hipertensión arterial o dislipidemia: Estudio piloto, comuna de Quellón, Chiloé. Revista chilena de cardiología, 37(2), 126-133. https://dx.doi.org/10.4067/S0718-85602018000200126
- https://adv-r.hadley.nz/index.html
- https://rstudio-education.github.io/hopr/
- https://bookdown.org/yihui/rmarkdown/