

# Taller R

## Sesion 4

Agosto 2022

# Índice

- Limpieza de datos (continuación: variables fecha, numérica y texto)
- `topper()`
- `unique()`
- `mutate()`
- `count()`

# Material de trabajo

Carpeta de drive:

<https://drive.google.com/drive/u/0/folders/14QDyRnqnLUjosCBIAzBqXL-qgIOMjNJL>

Base a descargar ile\_completa

Liga de zoom <https://us02web.zoom.us/j/83841692580>

ID de reunión: 838 4169 2580

# Que hemos visto:

- Estructura de datos
- Paqueterías o librerías
- Importar o subir archivos
- Función `unique()`, `str()`, `View()`
- Simbolo "\$" y pipe `%>%`
- Renombrar observaciones para limpiar datos
- función `count()`

# Empezamos con la limpieza de datos



# Antes de comenzar recuerda:

## Llamar librerías:

```
library(readxl)  
library(dplyr)
```

## Importar o cargar csv:

```
datos<-read.csv("C:/Users/Paola Viridiana/Desktop/ile_completa.csv", encoding = "U"
```

Recuerda que acá cambia tu ruta o ubicación de documenta y no olvides indicar el encoding UTF-8.

# Recordemos :)

## **unique()**

Extrae los valores unicos de una variable o columna.

## **toupper()**

Con esta función se convierte a mayúscula una columna o varias

## **mutate()**

La función mutate() sirve para crear nuevas variables.

## **case\_when**

Permite definir una variable, la cual toma un valor particular para cada condición establecida. En caso de no cumplir ninguna de las condiciones establecidas la variable tomara valor NA.

```
# con signo "$" se especifica una variable
datos$mes<- toupper(datos$mes)
```

# Variable year

Exploramos la variable year mostrando sus valores unicos

```
unique(datos$year)
```

```
## [1] 2016 2017 2018 2019 NA 2020
```

Parece todo bien... a excepción de los NA o valores perdidos

```
datos %>% count(year)
```

```
##   year      n
## 1 2016 18086
## 2 2017 17598
## 3 2018 17179
## 4 2019 15251
## 5 2020  6271
## 6  NA      21
```

Donde hay 21 NA's



# Variable fingresso

Esta variable hace referencia a la fecha de ingreso, consideremos tener registros del periodo 2016 a 2020, otras fechas se tomaran como valores perdidos o NA.

Acá otra forma de renombrar observaciones: Hay fechas que no tienen formato fecha (ymd) que son números enteros estas observaciones se pasaran a NA.

```
#Reemplazo valores no fecha a NA
datos$fingresso[datos$fingresso<=11112] <- NA_character_
datos$fingresso[datos$fingresso>43466] <- NA_character_
```

Se identifican dos fecha exactas que no entran en el periodo de análisis por lo cual se procede a reclasificarla como NA o valor perdido.

```
#Reemplazo fechas fuera de la periodicidad 2016-2020 a NA
datos$fingresso[datos$fingresso=="30/04/2048"] <- NA_character_
datos$fingresso[datos$fingresso=="25/11/2015"] <- NA_character_
```

Se observa una gran pérdida de información 64,571 NA :c

# Variable autorreferida

Esta variable especifica si la paciente es referida de otra unidad o acude directamente. Debe estar clasificada como sí = 1 y no = 0.

De todos los valores únicos se identifican los valores que pasan a NA y los pasan a un vector.

```
auto_no<-c(NA, "NE", "N/E")
```

Para remplazar se utiliza el vector nombrado "auto\_no"

```
#Con %in% le indicamos que busque dentro de la variable autoref los valores identificados
datos$AUTOREF_LIMPIA[datos$autoref %in% auto_no] <- NA_character_
```

```
# Los valores diff a auto_no que estén en la columna autoref pasan a 1
datos$AUTOREF_LIMPIA[datos$autoref != auto_no] <- 1
```

```
#Si hay observaciones = a NO se renombran con el valor 0
datos$AUTOREF_LIMPIA[datos$autoref == "NO"] <- 0
```

# Validar nueva variable autorreferida

Valores unicos de la nueva variable, solo se acepta 1, 0 y NA

```
unique(datos$AUTOREF_LIMPIA)
```

```
## [1] NA  "1" "0"
```

Conteo de la variable

```
datos %>% count(AUTOREF_LIMPIA)
```

```
##  AUTOREF_LIMPIA      n
## 1              0  7563
## 2              1 12947
## 3             <NA> 53896
```

# Variable edocivil\_descripcion: parte 1

## Paso 1: extraer valores unicos.

```
unique(datos$edocivil_descripcion)
```

## [1] "soltera"	"unión libre"	"casado(a)"
## [4] "divorciada"	"N/E"	"no sabe/ sin respuesta"
## [7] "casada (o)"	"separada"	"Soltera"
## [10] "Viuda"	"SOLTERA"	"VIUDA"
## [13] "viuda"	"S/INFORMACION"	"SE DESCONOCE"
## [16] "divociada"	"NO RESPONDE"	"UNION LIBRE"
## [19] "CASADA"	"SEPARADA"	"DIVORCIADA"
## [22] "soltero(a)"	"divorciado(a)"	"viudo"
## [25] "union libre"	"separado (a)"	NA
## [28] "Casada"	"Divorciada"	"Separada"
## [31] "UNIÓN LIBRE"	"SOLTERO/A"	"SEPARADO/A"
## [34] "DIVORCIADO/A"	"casada"	"SIN ESPECIFICAR"
## [37] "Sin especificar"		

# Variable edocivil\_descripcion: parte 2

Paso dos: pasar a mayúsculas.

```
# con signo "$" se especifica una variable
datos$edocivil_descripcion<- toupper(datos$edocivil_descripcion)
```

vemos valores unicos ahora que estan en mayúsculas

```
unique(datos$edocivil_descripcion)
```

```
## [1] "SOLTERA"           "UNIÓN LIBRE"       "CASADO(A)"
## [4] "DIVORCIADA"        "N/E"              "NO SABE/ SIN RESPUESTA"
## [7] "CASADA (O)"        "SEPARADA"         "VIUDA"
## [10] "S/INFORMACION"    "SE DESCONOCE"     "DIVOCIADA"
## [13] "NO RESPONDE"      "UNION LIBRE"      "CASADA"
## [16] "SOLTERO(A)"        "DIVORCIADO(A)"    "VIUDO"
## [19] "SEPARADO (A)"     NA                 "SOLTERO/A"
## [22] "SEPARADO/A"       "DIVORCIADO/A"     "SIN ESPECIFICAR"
```

# Variable edocivil\_descripcion: parte 3

Paso 3:reclasificar obsevaciones con case\_when cuando no se especifica alguna obsevación por default se pasa a NA o valor perdido.

```
datos<- datos %>%  
  mutate(  
    EDO_CIVIL_DESCRIPCION_LIMPIA = case_when(  
      edocivil_descripcion == "CASADO(A)" ~ "CASADA",  
      edocivil_descripcion == "SOLTERA" ~ "SOLTERA",  
      edocivil_descripcion == "UNIÓN LIBRE" ~ "UNIÓN LIBRE",  
      edocivil_descripcion == "DIVORCIADA" ~ "DIVORCIADA",  
      edocivil_descripcion == "CASADA (O)" ~ "CASADA",  
      edocivil_descripcion == "UNION LIBRE" ~ "UNIÓN LIBRE",  
      edocivil_descripcion == "SOLTERO(A)" ~ "SOLTERA",  
      edocivil_descripcion == "DIVORCIADO(A)" ~ "DIVORCIADA",  
      edocivil_descripcion == "VIUDO" ~ "VIUDA",  
      edocivil_descripcion == "SEPARADO (A)" ~ "SEPARADA",  
      edocivil_descripcion == "SOLTERO/A" ~ "SOLTERA",  
      edocivil_descripcion == "CASADA" ~ "CASADA",  
      edocivil_descripcion == "SOLTERO(A)" ~ "SOLTERA",  
      edocivil_descripcion == "SEPARADO/A" ~ "SEPARADA",  
      edocivil_descripcion == "DIVORCIADO/A" ~ "DIVORCIADA"  
    ))
```

Esta crea una nueva variable llamada EDO\_CIVIL\_DESCRIPCION\_LIMPIA se añadé a la base en la última columna.

# Variable edad

Se calcula las siguientes métricas:

```
summary(datos$edad)
```

```
##      Min. 1st Qu.  Median    Mean 3rd Qu.    Max.   NA's  
##    11.00   21.00   25.00   25.73   30.00   53.00     20
```

El promedio de edad de las mujeres es de 26, el mínimo 11 y el máximo 53 años.

Parece que esta variable esta limpia.

Se procede a caluclar rangos de edad por diversión:

```
datos<-datos %>%  
  mutate(RANGO_EDAD = case_when(edad < 17 ~ "MENOS DE 17 AÑOS",  
                                between(edad, 18, 25) ~ "18 a 25 AÑOS",  
                                between(edad, 26, 35) ~ "26 a 35 AÑOS",  
                                between(edad, 36, 45) ~ "36 a 45 AÑOS",  
                                edad > 46 ~ "MAYOR DE 46 AÑOS"))
```

# Conteo de variable edad

```
datos %>% count(RANGO_EDAD)
```

```
##           RANGO_EDAD      n
## 1      18 a 25 AÑOS 37980
## 2      26 a 35 AÑOS 26155
## 3      36 a 45 AÑOS  6504
## 4 MAYOR DE 46 AÑOS    46
## 5 MENOS DE 17 AÑOS  2201
## 6                <NA> 1520
```

Hay 1,520 datos perdidos.