



GUÍA DE LIMPIEZA DE DATOS

En muchas ocasiones es de vital importancia limpiar los datos y detectar valores que no existen o imposibles.

Para ellos te he preparado unas cuantas recetas para que puedas trabajar con tranquilidad una serie de acciones bastante comunes.

Hay más situaciones está claro.

Te invito que vayas creando tus propias recetas con la ayuda del foro stack overflow. No me voy a cansar de repetir que este foro es fundamental.

Te recuerdo las acciones que tienes disponibles:

- Imponer valores missing (NA)
- Borrar valores missing (NA)
- Borrar outliers
- Borrar duplicados
- Balancear casos – SMOTE

```
#library(tidyverse)
```

```
library(dlookr)
```

```
library(mlbench)
```

```
library(visdat)
```

```
library(plotly)
```

```
data(PimaIndiansDiabetes)
```

```
datos <- PimaIndiansDiabetes
```

```
#sustituir valores 0 por valores NA a lo largo de dataframe
```

```
datos <- datos %>% mutate(across(glucose:age, ~na_if(.,0)))
```

```
#podemos visualizar los valores NA dentro del dataframe
```

```
plot_na_pareto(datos, only_na = TRUE)
```

```
plot_na_intersect(datos)
```



GUÍA DE LIMPIEZA DE DATOS

```
#para poder ver en detalle las filas en donde se encuentran NA
```

```
vis_miss(datos) %>% ggplotly()
```

```
#imponer valores, vamos a comparar cual metodo funciona mejor
```

```
nombre <- "mass"
```

```
datos %>%
```

```
  impute_na(nombre, method = "mean") %>%
```

```
  plot()
```

```
datos %>%
```

```
  impute_na(nombre, method = "mice") %>%
```

```
  plot()
```

```
datos %>%
```

```
  impute_na(nombre, method = "rpart") %>%
```

```
  plot()
```

```
datos %>%
```

```
  impute_na(nombre, method = "knn") %>%
```

```
  plot()
```

```
#podemos imponer a todas las variables al mismo tiempo con missRanger
```

```
library(missRanger)
```

```
datos_impuestos <- missRanger(
```

```
  datos,
```

```
  formula = .~.,
```

```
  numb.trees = 1000,
```

```
  seed = 3
```

```
)
```

```
datos_impuestos %>% select(mass) %>% ggplot(aes(mass)) + geom_density()
```

```
datos %>% select(mass) %>% ggplot(aes(mass)) + geom_density()
```

```
#Borrar valores missing (NA)
```



```
datos_completos <- datos %>% filter(complete.cases(.))
```

```
datos_completos2 <- datos %>% na.omit
```

```
library(tidyr)
```

```
datos_completos3 <- datos %>% drop_na()
```

```
#Borrar outliers
```

```
library(flextable)
```

```
#Podemos diagnosticar los outliers
```

```
diagnose_outlier(datos_impuestos) %>% flextable()
```

```
outliers <- find_outliers(datos_impuestos, index = F)
```

```
datos_impuestos %>%
```

```
  select(outliers) %>%
```

```
  plot_outlier()
```

```
nombre <- "pressure"
```

```
datos_impuestos %>%
```

```
  impute_outlier(nombre, method = "mean") %>%
```

```
  plot()
```

```
datos_impuestos %>%
```

```
  impute_outlier(nombre, method = "median") %>%
```

```
  plot()
```

```
datos_impuestos %>%
```

```
  impute_outlier(nombre, method = "mode") %>%
```

```
  plot()
```

```
datos_impuestos %>%
```

```
  impute_outlier(nombre, method = "capping", cap_ntiles = c(0.003, 0.997)) %>%
```

```
  plot()
```

```
#Balancear casos – SMOTE
```

```
#Para usar la librería DMwR se tiene que usar una versión antigua de R en mi caso utilice  
#la 4.0.0 y previo instalar los paquetes.
```

```
install.packages("mlbench")
```

```
library(mlbench)
```

```
library(DMwR)
```

```
# load the dataset
```

```
data(PimaIndiansDiabetes)
```

```
datos <- PimaIndiansDiabetes
```

```
# display count of instances of each class (unbalanced)
```

```
table(datos$diabetes)
```

```
# use SMOTE to created a "more balance" version of the dataset
```

```
datos_balanceados <- SMOTE(diabetes~., datos, perc.over=400, perc.under=100)
```

```
table(datos_balaceados$diabetes)
```

Utiliza estas 5 recetas para eliminar casos NA, identificar casos que quieres eliminar y balancear la tabla de datos.

¡Seguimos!