

# Análisis del medio abiótico

Oscar Inostroza-Michael

2024-04-15

Este tutorial está diseñado para aprender a realizar análisis básicos sobre datos del *medio abiótico* utilizando nuestro programa de R **limnoHOLON**. Los análisis implementados a la fecha son:

1. Tabla de estadísticos descriptivos.
2. Gráfico barra de concentración/niveles de parámetros de laboratorio analizados.
3. Correlograma (*i.e.* gráfico de correlaciones pareadas).
4. Gráfico PCA.
5. Gráfico de granulometría (solo para sedimentos).

## Datos de entrada

Como en cualquier otro tipo de análisis, los datos de entrada son **fundamentales**. Si no representan la información pretendida adecuadamente, los resultados y conclusiones posteriores podrían ser erróneos. Como regla general, los datos deben cumplir con ciertas condiciones mínimas de formato:

1. Cada entrada de datos implica un gasto considerable de recursos (tiempo, plata, etc). Así que, hay que asegurarse de que cada entrada represente un valor definido y correcto. Las celdas vacías, típicamente son interpretados como casos NA (*Not available*). Sin embargo, para quien hizo la base datos esa celda no la relleno por ser una valor = 0. Por ejemplo:

```
vec1 <- c(1, 2, 3, 4, 5) # vector con los valores reales
vec2 <- c(1, 2, 3, 4, 0) # vector con la última entrada con NA
vec3 <- c(1, 2, 3, 4, NA) # vector con la última entrada con 0
avg1 <- mean(vec1)
avg2 <- mean(vec2)
avg3 <- mean(vec3)
print(avg1)
#> [1] 3
print(avg2)
#> [1] 2
print(avg3)
#> [1] NA
```

El vector vec3, al tener un **NA** en su ultima posición, se produce un *error* y entrega un **NA** como resultado. Dicho error puede ser fácilmente *corregido* con:

```
avg3 <- mean(vec3, na.rm = TRUE)
print(avg3)
#> [1] 2.5
```

Pero el resultado no es el correcto.

2. Formato *long*; es decir, una variable por columna y en las filas cada caso.
3. Nombres de las variables concisos, no repetidos, sin tildes, espacios ni caracteres especiales. Generalmente, se usa notaciones tales como : *camelCase* (nombreVariable), *snakecase* (nombre\_variable), *flatcase*

(nombrevARIABLE).

4. Generalmente y para no incurrir en errores involuntarios en la lectura de datos, se recomiendan los formatos **.csv** ( *comma separated values* ) o **.tsv** ( *tab separtared values*). Este último es más sencillo ya que solo hay que copiar la hoja de Excel y pegarla en un documento de texto plano. Copiar y pegar en un documento de texto plano tiene la ventaja adicional de ver si se están colando caracteres inesperados.

## Análisis del medio abiótico

### Tabla de estadísticos descriptivos

Habiendo revisado los puntos anteriores, proseguiremos con el análisis del medio abiótico. Para esto utilizaremos los datos de XXX junto a nuestro paquete de R

```
# instalación Para que funcione, se necesitan instalar previamente los
# paquetes: RColorBrewer, corrplot, ggpubr, grDevices, Knitr, patchwork, rlang,
# rstatix, scales, tidyverse, vegan (ejemplo :
# install.packages('RColorBrewer')) Una vez listo, instalamos y cargamos
# limnoHOLON devtools::install_github('oscarIM/limnoHOLON')
library(tidyverse)
#> -- Attaching core tidyverse packages ----- tidyverse 2.0.0 --
#> v dplyr      1.1.4      v readr      2.1.5
#> v forcats    1.0.0      v stringr   1.5.1
#> v ggplot2     3.4.4      v tibble    3.2.1
#> v lubridate  1.9.3      v tidyr     1.3.1
#> v purrr      1.0.2
#> -- Conflicts ----- tidyverse_conflicts() --
#> x dplyr::filter() masks stats::filter()
#> x dplyr::lag()    masks stats::lag()
#> i Use the conflicted package (<http://conflicted.r-lib.org/>) to force all conflicts to become errors
library(scales)
#>
#> Attaching package: 'scales'
#>
#> The following object is masked from 'package:purrr':
#>
#>   discard
#>
#> The following object is masked from 'package:readr':
#>
#>   col_factor
library(RColorBrewer)
library(grDevices)
library(vegan)
#> Loading required package: permute
#> Loading required package: lattice
#> This is vegan 2.6-4
library(limnoHOLON)
# Para ver las funciones con sus páginas de ayuda
help(package = "limnoHOLON")

data <- readr::read_tsv("data_agua_VEN.tsv")
#> Rows: 450 Columns: 6
#> -- Column specification -----
#> Delimiter: "\t"
```

```

#> chr (5): Sitio, zonas, nombre_par, Unidad, Param
#> dbl (1): Valor
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
head(data)
#> # A tibble: 6 x 6
#>   Sitio zonas nombre_par      Unidad Valor Param
#>   <chr> <chr> <chr>      <chr>   <dbl> <chr>
#> 1 P-1   Q1    Aceites y Grasas mg/L    0.992 AyG
#> 2 P-10  Q3    Aceites y Grasas mg/L    2.18  AyG
#> 3 P-11  Q3    Aceites y Grasas mg/L    1.89  AyG
#> 4 P-12  L     Aceites y Grasas mg/L    1.15  AyG
#> 5 P-13  L     Aceites y Grasas mg/L    2.3    AyG
#> 6 P-14  L     Aceites y Grasas mg/L    0.5    AyG
col_pars <- "Param"
data_pars <- readr::read_tsv("tabla_pars_master.tsv")
#> Rows: 52 Columns: 3
#> -- Column specification -----
#> Delimiter: "\t"
#> chr (3): nombre_par, Param, cats_pars
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
col_valor <- "Valor"
matriz <- "agua"
round <- 4
fn_stats(data = data, col_pars = col_pars, col_valor = col_valor, data_pars = data_pars,
          matriz = matriz, round = round)
#>
#>
#> Table: Tabla parámetros
#>
#> | Sigla | Nobs | min | max | prom | desvest | cv% |
#> |-----|-----|-----|-----|-----|-----|-----|
#> | Cond | 25 | 67.900 | 306.000 | 153.1760 | 81.8245 | 53% |
#> | Redox | 25 | 1.222 | 63.490 | 27.9412 | 14.3364 | 51% |
#> | Temp | 25 | 11.670 | 19.660 | 14.6456 | 2.9690 | 20% |
#> | Turb | 25 | 1.610 | 28.000 | 9.0844 | 8.5858 | 95% |
#> | pH | 25 | 5.822 | 8.603 | 7.3503 | 0.5506 | 7% |
#> | DBO5 | 25 | 2.000 | 4.900 | 2.5880 | 0.7412 | 29% |
#> | DQO | 25 | 2.400 | 103.000 | 11.0400 | 19.3979 | 176% |
#> | Odis | 25 | 1.454 | 9.176 | 7.4278 | 1.9638 | 26% |
#> | SST | 25 | 5.000 | 66.000 | 20.2400 | 18.8199 | 93% |
#> | NH4 | 25 | 0.100 | 1.000 | 0.2144 | 0.1794 | 84% |
#> | NO2 | 25 | 0.030 | 0.030 | 0.0300 | 0.0000 | 0% |
#> | NO3 | 25 | 0.300 | 2.000 | 0.8720 | 0.4730 | 54% |
#> | NT | 25 | 0.720 | 3.490 | 1.5064 | 0.6841 | 45% |
#> | NTK | 25 | 0.150 | 2.690 | 0.6332 | 0.5374 | 85% |
#> | PT | 25 | 0.003 | 0.183 | 0.0228 | 0.0438 | 192% |
#> | AyG | 25 | 0.500 | 2.460 | 1.2993 | 0.6073 | 47% |
#> | ColFec | 25 | 1.800 | 20.000 | 3.6360 | 5.1304 | 141% |
#> | ColTot | 25 | 1.800 | 45.000 | 5.4640 | 9.6907 | 177% |

```

## Gráfico barra de concentración/niveles de parámetros

```
# Sin grupos
data <- readr::read_tsv("data_agua_VEN.tsv")
#> Rows: 450 Columns: 6
#> -- Column specification -----
#> Delimiter: "\t"
#> chr (5): Sitio, zonas, nombre_par, Unidad, Param
#> dbl (1): Valor
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
col_pars <- "Param"
col_sitio <- "Sitio"
col_valor <- "Valor"
col_unidad <- "Unidad"
matriz <- "agua"
code_sitio <- "P-"
ord_sitio <- "asc" # 'desc o asc'
data_pars <- readr::read_tsv("tabla_pars_master.tsv")
#> Rows: 52 Columns: 3
#> -- Column specification -----
#> Delimiter: "\t"
#> chr (3): nombre_par, Param, cats_pars
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
width <- 12
height <- 11
fn_plot_bar_abiotic(data = data, col_pars = col_pars, col_sitio = col_sitio, col_valor = col_valor,
  aspect_ratio = 1, col_unidad = col_unidad, code_sitio = code_sitio, matriz = matriz,
  data_pars = data_pars, ord_sitio = ord_sitio, width = width, height = height)
```

```
# Con un grupo (generalmente son zonas para agrupar puntos, campañas, etc.) en
# donde los niveles de este se ordenen de forma alfanumérica.
data <- readr::read_tsv("data_agua_VEN.tsv")
#> Rows: 450 Columns: 6
#> -- Column specification -----
#> Delimiter: "\t"
#> chr (5): Sitio, zonas, nombre_par, Unidad, Param
#> dbl (1): Valor
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
col_pars <- "Param"
col_sitio <- "Sitio"
col_valor <- "Valor"
col_unidad <- "Unidad"
col_grupo <- "zonas"
matriz <- "agua"
code_sitio <- "P-"
ord_sitio <- "asc" # 'desc o asc'
data_pars <- readr::read_tsv("tabla_pars_master.tsv")
#> Rows: 52 Columns: 3
```

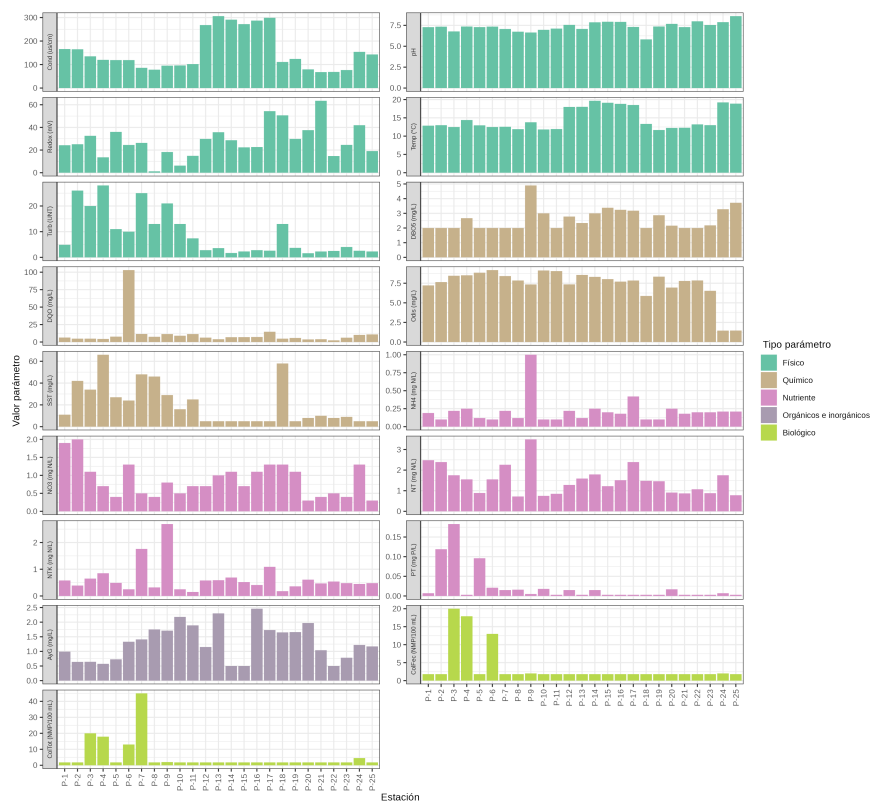


Figure 1: Gráfico de concentración/nivel de parámetros a lo largo de las estaciones de muestreo.

```

#> -- Column specification -----
#> Delimiter: "\t"
#> chr (3): nombre_par, Param, cats_pars
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
width <- 12
height <- 11
fn_plot_bar_abiotic(data = data, col_pars = col_pars, col_sitio = col_sitio, col_valor = col_valor,
  col_grupo = col_grupo, aspect_ratio = 2, col_unidad = col_unidad, code_sitio = code_sitio,
  matriz = matriz, data_pars = data_pars, ord_sitio = ord_sitio, width = width,
  height = height)

```



Figure 2: Gráfico de concentración/nivel de parámetros a lo largo de las estaciones de muestreo por grupo.

### Gráfico de correlaciones pareadas (correlograma)

```

data <- readr::read_tsv("data_agua_VEN.tsv")
#> Rows: 450 Columns: 6
#> -- Column specification -----
#> Delimiter: "\t"
#> chr (5): Sitio, zonas, nombre_par, Unidad, Param
#> dbl (1): Valor
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
col_pars <- "Param"

```

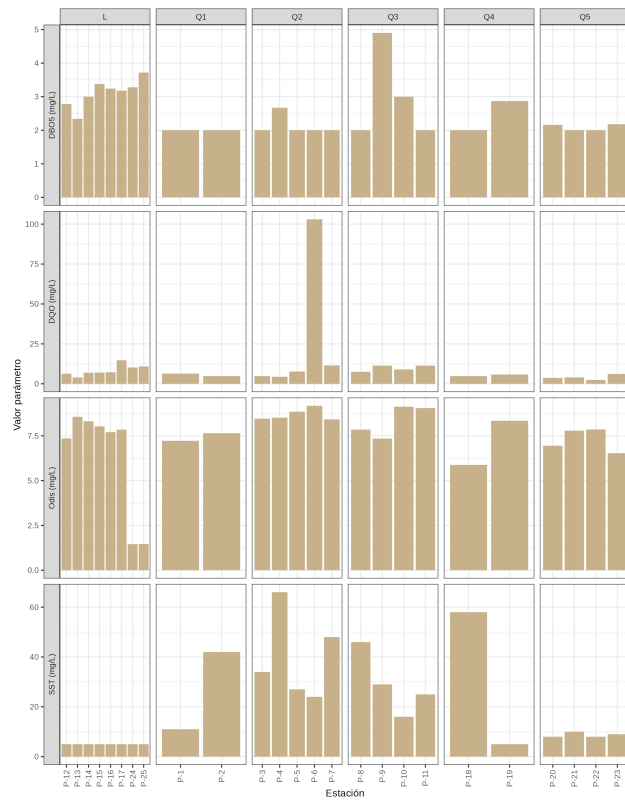


Figure 3: Gráfico de concentración/nivel de parámetros a lo largo de las estaciones de muestreo por grupo.

```

col_sitio <- "Sitio"
col_valor <- "Valor"
matriz <- "agua"
code_sitio <- "P-"
data_pars <- readr::read_tsv("tabla_pars_master.tsv")
#> Rows: 52 Columns: 3
#> -- Column specification -----
#> Delimiter: "\t"
#> chr (3): nombre_par, Param, cats_pars
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
width <- 6
height <- 6
fn_plot_correlogram(data = data, col_pars = col_pars, col_sitio = col_sitio, matriz = matriz,
  code_sitio = code_sitio, data_pars = data_pars, width = width, height = width)
#> pdf
#> 2

```

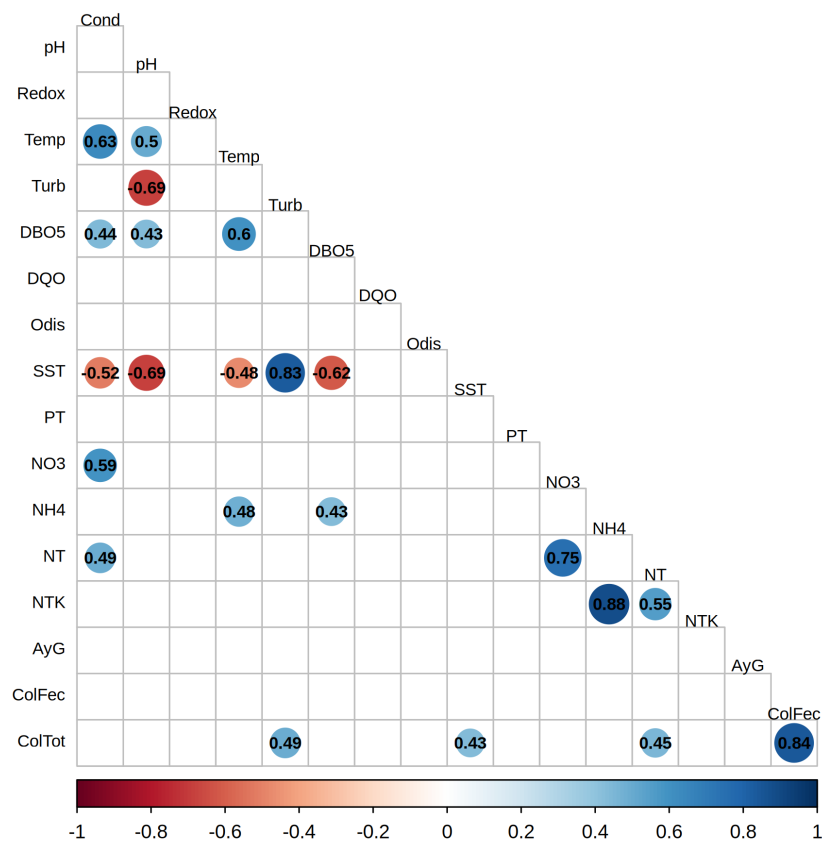


Figure 4: Gráfico de correlaciones pareadas.

### Gráfico de PCA.

```

# figura sin factor de agrupamiento
data <- readr::read_tsv("data_agua_VEN.tsv")

```



```

#> Rows: 450 Columns: 6
#> -- Column specification -----
#> Delimiter: "\t"
#> chr (5): Sitio, zonas, nombre_par, Unidad, Param
#> dbl (1): Valor
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
col_pars <- "Param"
col_sitio <- "Sitio"
col_valor <- "Valor"
matriz <- "agua"
data_pars <- readr::read_tsv("tabla_pars_master.tsv")
#> Rows: 52 Columns: 3
#> -- Column specification -----
#> Delimiter: "\t"
#> chr (3): nombre_par, Param, cats_pars
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
width <- 8
height <- 6
fn_plot_pca(data = data, col_pars = col_pars, col_sitio = col_sitio, col_valor = col_valor,
  data_pars = data_pars, width = width, height = height, matriz = matriz)

```

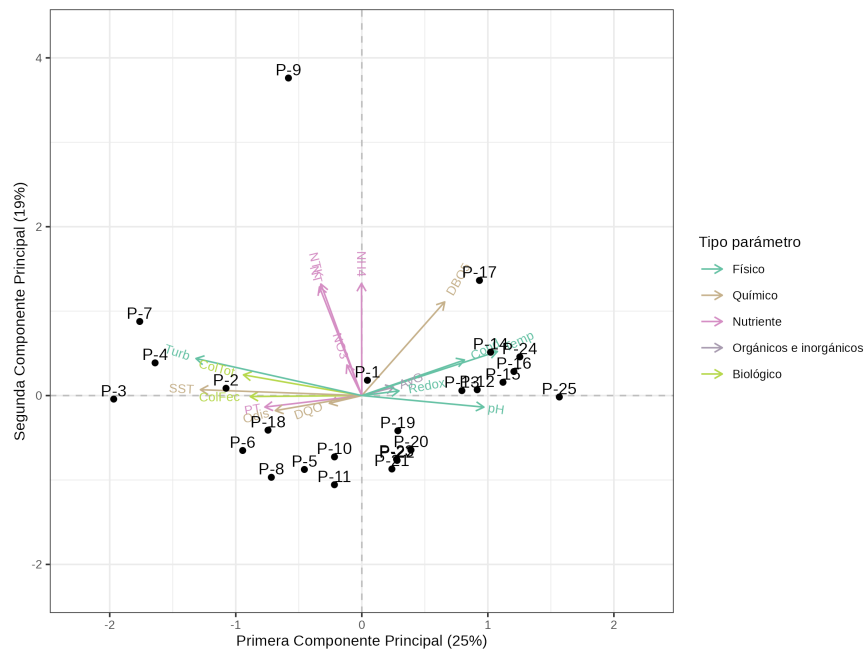


Figure 5: Gráfico PCA sin factor de agrupamiento

```

# figura con 1 factor de agrupamiento#
data <- readr::read_tsv("data_agua_VEN.tsv")
#> Rows: 450 Columns: 6
#> -- Column specification -----
#> Delimiter: "\t"

```

```

#> chr (5): Sitio, zonas, nombre_par, Unidad, Param
#> dbl (1): Valor
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
col_pars <- "Param"
col_sitio <- "Sitio"
col_valor <- "Valor"
matriz <- "agua"
code_sitio <- "P-"
data_pars <- readr::read_tsv("tabla_pars_master.tsv")
#> Rows: 52 Columns: 3
#> -- Column specification -----
#> Delimiter: "\t"
#> chr (3): nombre_par, Param, cats_pars
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
col_grupo <- "zonas"
ord_grupo <- c("L", "Q1", "Q2", "Q3", "Q4", "Q5")
width <- 7
height <- 5
dist <- "euc"
fn_plot_pca(data = data, col_pars = col_pars, col_sitio = col_sitio, col_valor = col_valor,
  data_pars = data_pars, width = width, height = height, matriz = matriz, col_grupo = col_grupo,
  ord_grupo = ord_grupo, dist = dist)
#> Too few points to calculate an ellipse
#> Too few points to calculate an ellipse

```

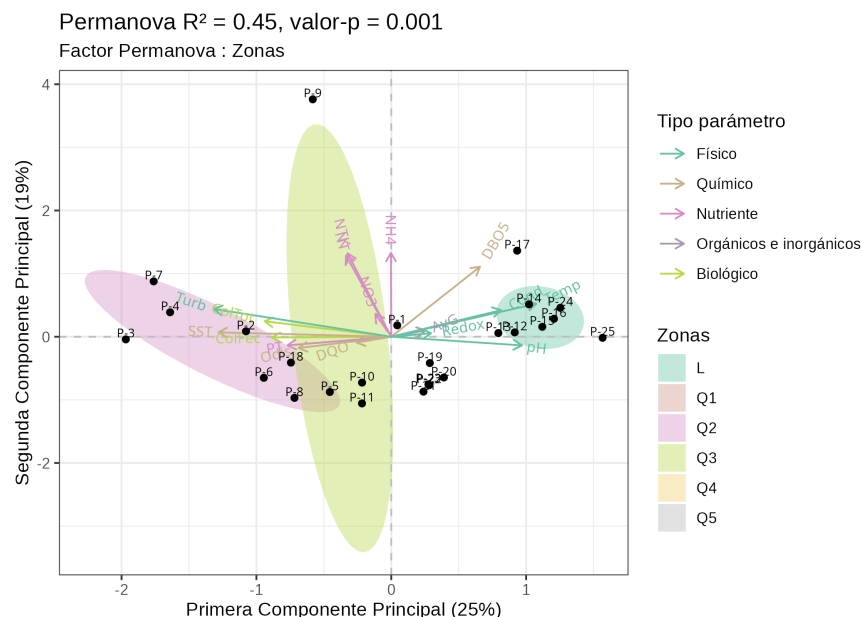


Figure 6: Gráfico PCA sin factor de agrupamiento

## Gráfico de granulometría.

Obviamente, este gráfico tiene sentido solo si se usa sobre datos de sedimentos, y que contengan las fracciones : “LIM”, “AMF”, “AF”, “AM”, “AG”, “AMG”, “GRAN”

```
# gráfico sin factor o grupo
data <- readr::read_tsv("data_sedimentos_VEN.tsv")
#> Rows: 300 Columns: 6
#> -- Column specification -----
#> Delimiter: "\t"
#> chr (5): Sitio, zonas, nombre_par, Unidad, Param
#> dbl (1): Valor
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
col_pars <- "Param"
col_sitio <- "Sitio"
col_valor <- "Valor"
code_sitio <- "P-"
ord_sitio <- "asc" # 'desc o asc'
width <- 8
height <- 7
fn_plot_granulometria(data = data, col_pars = col_pars, col_sitio = col_sitio, col_valor = col_valor,
  ord_sitio = ord_sitio, width = width, height = height, aspect_ratio = 2, code_sitio = code_sitio)
```

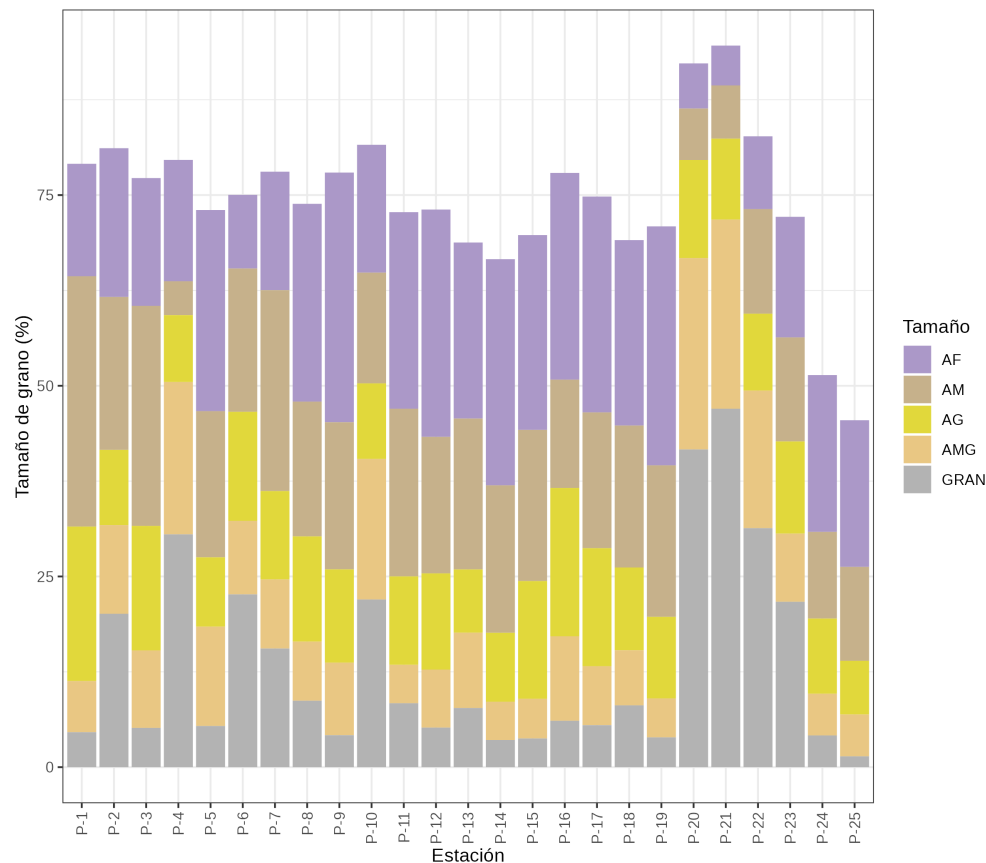


Figure 7: Gráfico de granulometría sin factor de agrupamiento

```

# gráfico con 1 factor o grupo
data <- readr::read_tsv("data_sedimentos_VEN.tsv")
#> Rows: 300 Columns: 6
#> -- Column specification -----
#> Delimiter: "\t"
#> chr (5): Sitio, zonas, nombre_par, Unidad, Param
#> dbl (1): Valor
#>
#> i Use `spec()` to retrieve the full column specification for this data.
#> i Specify the column types or set `show_col_types = FALSE` to quiet this message.
col_pars <- "Param"
col_sitio <- "Sitio"
col_valor <- "Valor"
ord_sitio <- "asc" # 'desc o asc'
width <- 8
height <- 7
col_grupo <- "zonas"
code_sitio <- "P-"
# ord_grupo <- c('Q1', 'Q2', 'Q3', 'Q4', 'Q5', 'L')
fn_plot_granulometria(data = data, col_pars = col_pars, col_sitio = col_sitio, col_valor = col_valor,
  ord_sitio = ord_sitio, width = width, height = height, col_grupo = col_grupo,
  ord_grupo = NULL, code_sitio = code_sitio, aspect_ratio = 2)

```

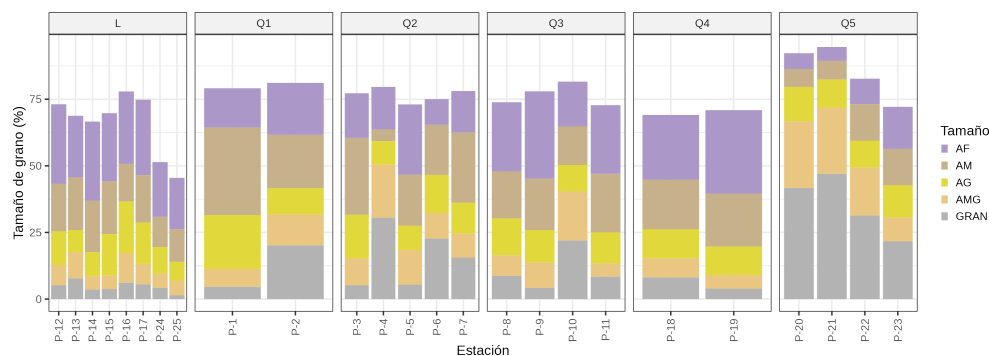


Figure 8: Gráfico de granulometría con factor de agrupamiento