Trabajo Práctico Nº 3

ACP - Clustering

Presentado en la fecha: 09/09/2023

Hecho por: Huarca Brian

Nicolas Benitez

Facundo Rodriguez

Derlis Walter Hodge

Contents

Resumen **3**

Sumario **4**

0.1 Sumario 4

Objetivo **5**

Desarrollo **6**

1 Dataset **6**

1.1 Informacion del dataset 6

1.2 Diccionario de datos 8

2 Dataset **11**

2.1 Analisis de datos 11

2.2 Summary de los datos 12

2.3 Deteccion de outliers 13

2.4 Manejo de Outliers 15

2.5 Matriz de correlación 17

3 ACP **18**

3.1 Prueba de esfericidad Bartlett 18

3.2 Prueba KMO 18

3.3 Importancia de los componentes 19

3.4 Grafico de Sedimentacion 20

3.5 Biplot 21

4 Clustering **22**

4.1 Clustering jerárquico 22

4.2 Clustering no jerárquico : K-means 23

Conclusión **25**

Anexo **26**

Resumen

El presente informe detallamos el Analisis de componentes sobre un conjunto de variables correlacionadas y simplificar la cantidad de variables en un nuevo conjunto no correlacionado.
Tambien se detalla el uso de Clusters o Agrupamiento para la clasificacion de individuos en grupos homogeneos.

Sumario

0.1 Sumario

- Preparación del dataset (Valores Null, Outliers).
- Análisis de la relación entre variables(Correlograma).
- Analisis de los Test Bartlett y KMO
- Análisis de las componentes principales
- Analisis de Linkage
- Analaisis de Clusters
- Exposicion de graficos como ejemplo

Objetivo

El informe tiene por objetivo en primer lugar a partir de un analisis de componentes principlaes(ACP) reducir la dimension que describen la informacion una gran cantidad de variables en una cantidad mas chica, analizar cuantas componentes explican mejor la infomacion de los datos. En segundo realizar un analisis de agrupamiento(clustering) para ver de que forma se agrupan los datos

Dataset

1.1 Informacion del dataset

Este docuemento presenta el monitoreo de la calidad del agua en el Rio de la Plata en la campaña otoño 2023 Se muestran los parametros de la calidad del agua de la Red de Intercambio de informacion de los gobiernos locales (RIIGLO).
El Centro de Información Ambiental (CIAM), creado por Resolución MAyDS Nº161/2020 pone a disposición de la ciudadanía, información generada en el ámbito del Estado Nacional, con aportes de otras instituciones, la academia, la sociedad civil y el sector privado. El CIAM dispone de un plataforma, Sistema integrado de informacion Ambiental (SInIA) a la que se puede acceder a datos, estadisticas e indicadores ambientales

Parámetros		Valor de Referencia (Uso recreativo con contacto directo)	
		Valor	Norma
Físico- químicos	Oxígeno Disuelto*	>5 mg/l	Res. ACUMAR 46/2017
	pH*	6,5 - 9	Res ADA 42
	Temperatura*	<35	Res. ACUMAR 46/2017
	Turbidez	100	Res ADA 42
Organolépticos	Color	No perceptible	Res ADA 42
	Olor	No perceptible	Res ADA 42
	Materiales flotando y espumas no naturales	No se observan	Res ADA 42
Bacteriológico	Coliformes Fecales	150 UFC/100 ml	Res. ACUMAR 46/2017
	Escherichia coli	126 UFC/100 ml	Res. ACUMAR 46/2017
	Enterococos	33 UFC/100 ml	Res ADA 42
Nutrientes (Eutrofización)	Nitratos (NO3-)	125 mg/l	Res ADA 42
	Amonio (NH4+)	0,5 mg/l	Res ADA 42
	Fósforo Total	0,025 mg/l	Res ADA 42
	Fosfatos ($PO_4^{\equiv}$)	X	
	Clorofila 'a'	50 ug/l	Res ADA 42
	Microcistina	10 ug/l	Res ADA 42
Mat. Org.	DBO_5	10	Res ADA 42
	DQO	X	
Derivados del Petróleo	Hidrocarburos derivados del Petróleo	< 50 ug/l	Res. ACUMAR 46/2017
Metales Pesados	Cromo Total	50 ug/l	Res. ACUMAR 46/2017
	Cadmio Total	5 ug/l	Res. ACUMAR 46/2017

Figure 1: Diccionario de datos

Fuente: [Ministerio de Ambiente y desarrollo sostenible]

1.2 Diccionario de datos

- **tem agua** = temperatura del agua
- **tem aire** = temperatura del aire
- **od** = oxigeno disuelto
- **ph** = medida que indica la acidez o la alacalinidad del agua
- **olores** = flag de presencia/ausencia
- **color** = flag de presencia/ausencia
- **espumas** = flag de presencia/ausencia (Espumas no naturales)
- **mat susp** = flag de presencia/ausencia (Materiales flotando)
- **colif fecales ufc 100ml** = Coliformes Fecales (Son contaminantes comunes del tracto gastrointestinal tanto del hombre como de los animales de sangre caliente) - GRUPO DE BACTERIAS
- **escher coli ufc 100ml** = Escherichia coli (Es un tipo de bacteria que se encuentra comúnmente en los intestinos de animales y seres humanos) - TIPO DE BACTERIAS
- **enteroc ufc 100ml** = Enterococos: Es un indicador bacteriológico para aguas marinas o salobres, y que son más resistentes, y tienen mejor relación con las enfermedades gastrointestinales, respiratorias y dermatológicas
- **nitrato mg l** = NUTRIENTES: El nitrato se acumula en las cuencas hidrográficas agrícolas donde los agricultores esparcen fertilizantes inorgánicos y abono animal en las tierras de cultivo. El nitrógeno que no es absorbido por los cultivos puede filtrarse a través del suelo al agua subterránea y luego fluir a áreas de recarga o pozos privados.
- **nh4 mg l** = EUTROFIZACION: Amonio: La presencia de niveles altos de amonio puede comprometer la eficacia de la desinfección del agua o provocar fallos en la eliminación del manganeso en los filtros, lo que puede dar problemas de sabor y olor en el agua

- **p total l mg l** = Fosforo total: tiene como fuente principal el uso de fertilizantes agrícolas, aunque proviene también de la erosión del suelo y la materia orgánica en descomposición que descargan industrias, urbes y granjas de animales domésticos.
- **fosf ortofos mg l** = Fosfato: Los fosfatos se añaden a los detergentes para contrarrestar la dureza del agua y maximizar la eficacia de la limpieza. Pero cuando llegan a los lagos y ríos contribuyen a la proliferación de algas que matan a los peces al privarles de oxígeno en el agua.
- **dbo mg l** = CANTIDAD DE OXIGENO NECESARIO PARA DESCOMPONER QUIMICAMENTE LA MATERIA A TRAVES DE MICROORGANISMOS
- **dqo mg l** = CANTIDAD DE OXIGENO NECESARIO PARA DESCOMPONER QUIMICAMENTE LA MATERIA A TRAVES DE MEDIOS QUIMICOS (indica la cantidad de oxígeno necesaria para la oxidación de todas las sustancias orgánicas del agua)
- **turbiedad ntu** = Mita la claridad del agua: Es la unidad en la que se mide la turbidez de un fluido o la presencia de partículas en suspensión en el agua, cuantos más sólidos en suspensión haya en el agua, más sucia parecerá esta y más alta será la turbidez.
- **hidr deriv petr ug l** = Hidrocarburos derivados del Petróleo
- **cr total mg l** = Cromo total: Metal super contaminante
- **cd total mg l** = El cadmio contamina el agua sobre todo por los vertidos de aguas residuales sin tratar de industrias como las del acabado de metales, la electrónica, las aleaciones de hierro y la producción de hierro y zinc, la fabricación de pigmentos (pinturas y colorantes), de baterías (cadmio, níquel)
- **clorofila a ug l** = la clorofila en el agua es un indicador de la actividad fotosintética de los organismos acuáticos y es importante para comprender y monitorear la salud de los ecosistemas acuáticos.
- **microcistina ug l** = Toxinas de Algas: Las microcistinas son metabolitos secundarios que normalmente se encuentran en el interior de la célula. Sin embargo, cuando la toxina es liberada, normalmente por lisis celular, el agua queda contaminada y su consumo es nocivo no solo para el ser humano si no también para los animales.

- **ica** = ica
- **calidad de agua** = detalle de calidad de agua

Dataset

2.1 Analisis de datos

Verificamos que no existiesen valores Null (desconocidos) en el dataset.

Analizando el data set nos damos cuenta, teniendo en cuenta si se quiere hacer una regresion, nuestra variable que podria ser predictora "ICA" tiene un total de 17 filas con N/A teniendo todos los datos para calcular la calidad del agua. Al dejar la columna ICA y verificar la cantidad de N/A nos quedan un total de 48 filas, sin esa columna nos queda una cantidad restante de 53.

2.2 Summary de los datos

```
> summary(agc_rio_plata_23_out)
 escher_coli_ufc    enteroc_ufc         amonio         microcistina    turbiedad_ntu     clorofila_ug
 Min.   :    1.5   Min.   :   1.0   Min.   : 0.050   Min.   : 30.00   Min.   :  2.90   Min.   :  0.10
 1st Qu.:  100.0   1st Qu.:  50.0   1st Qu.: 0.130   1st Qu.: 30.00   1st Qu.: 21.00   1st Qu.:  0.91
 Median :  400.0   Median : 120.0   Median : 0.510   Median : 30.00   Median : 27.00   Median :  8.57
 Mean   : 1849.2   Mean   : 629.5   Mean   : 1.514   Mean   : 38.47   Mean   : 40.47   Mean   : 31.48
 3rd Qu.: 1100.0   3rd Qu.: 550.0   3rd Qu.: 0.990   3rd Qu.: 30.00   3rd Qu.: 45.00   3rd Qu.: 25.58
 Max.   :32000.0   Max.   :9300.0   Max.   :18.000   Max.   :230.00   Max.   :432.00   Max.   :740.93
>
```

Figure 2: Summary del data set

Podemos observar que en algunas variables nuestro valor maximo se aleja de forma poco o muy considerable entre el valor maximo y el 3er cuartil.

2.3 Deteccion de outliers

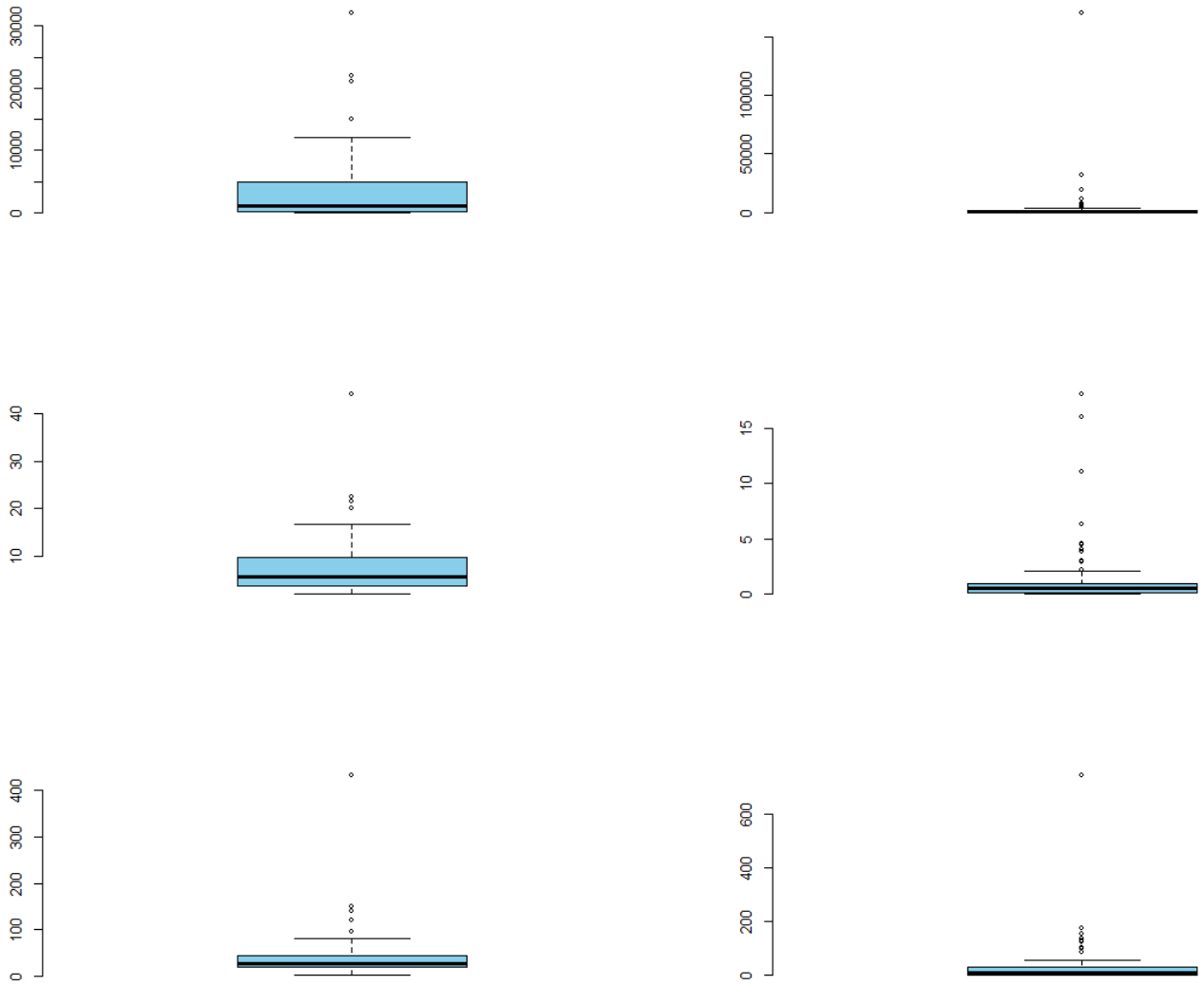

Figure 3: Boxplot 1

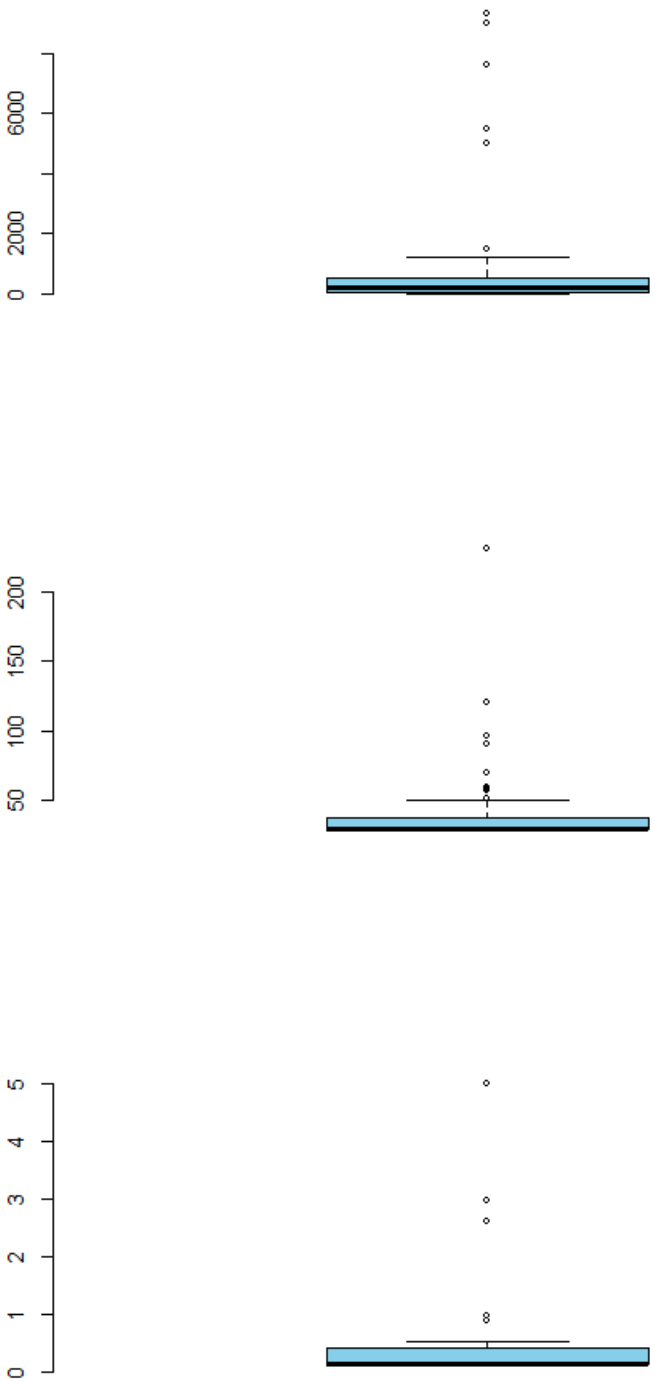

Figure 4: Boxplot 2

2.4 Manejo de Outliers

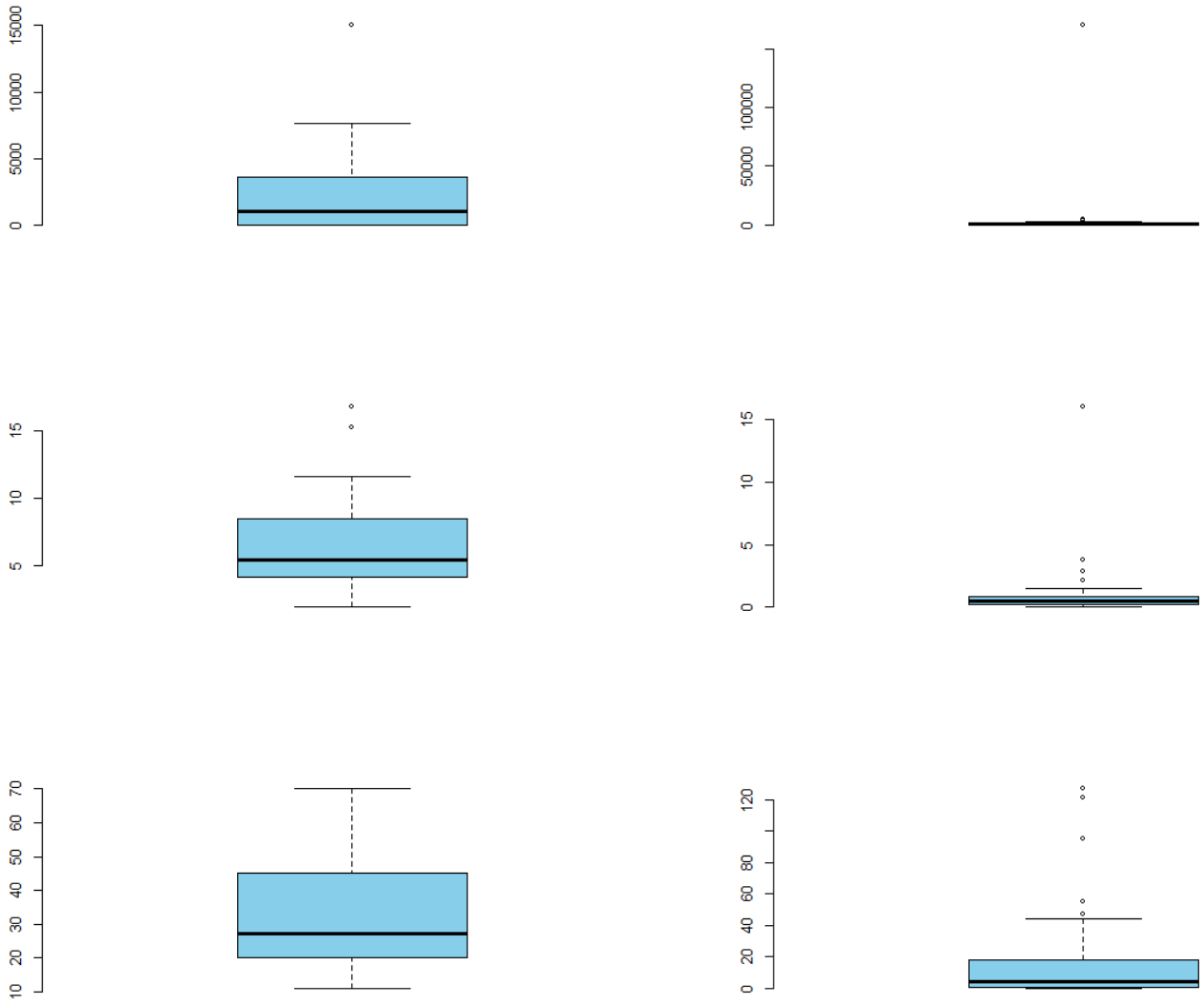

Figure 5: summary de las componentes

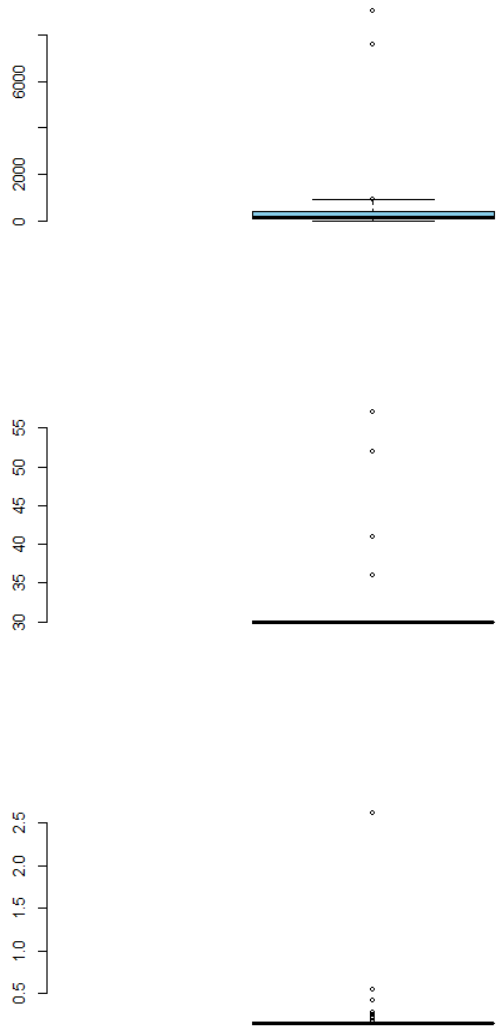

Figure 6: summary de las componentes

2.5 Matriz de correlación

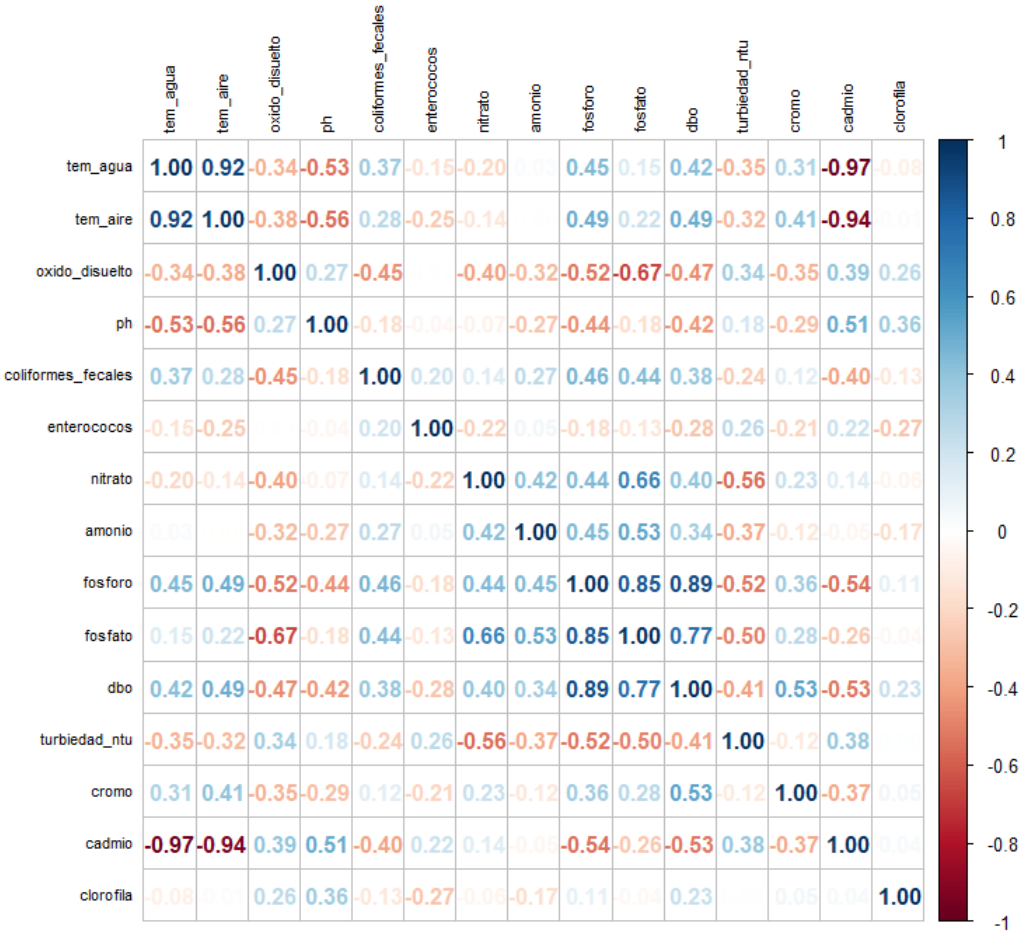

Figure 7: Matriz de correlación entre variables

De la matriz podemos observar que nuestras variables mas influyentes son la temperaturas que tienen una alta correlacion con el indice de oxigeno disuelto en el agua y el indice de acides pero de forma inversa. A su vez se observa que los indices de limpieza de quimicos se encuentran altamente correlacionado con oxido disuelto, nitrado, fosforo, y dbo (gran parte de las variables tanto de forma positiva como negativa).

ACP

3.1 Prueba de esfericidad Bartlett

La prueba de bartlett nos indica un p-value chico por lo que podemos rechazar la hipotesis nula y es apto para hacer una analisis de componentes principales

```
$chisq
[1] 376.88

$p.value
[1] 2.647443e-32

$df
[1] 105
```

Figure 8: Bartlett

3.2 Prueba KMO

Indica si los datos son adecuados para aplicar un ACP. EL valor cercano a 1 indica que es aceptable para realizar el analisis. En este caso 0.69

```
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = cor(mca_rlp_esc))
Overall MSA =  0.69
MSA for each item =
   tem_agua    tem_aire oxido_disuelto          ph coliformes_fecales  enterococos     nitrato
       0.81        0.81           0.80        0.45              0.83         0.57        0.61
    amonio     fosforo     fosfato         dbo turbiedad_ntu       cromo      cadmio
       0.81        0.75           0.59        0.76              0.76         0.66        0.75
 clorofila
       0.21
```

Figure 9: KMO

3.3 Importancia de los componentes

El siguiente grafico indica el porcentaje de la proporcion y el acumulado que cada variable describe mejor la informacion de los datos.
Hasta la componente 3 tenemos el 67% de los datos explicados

```
Importance of components:
                          PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8    PC9    PC10    PC11    PC12    PC13    PC14
Standard deviation     2.4087 1.6162 1.3222 1.02593 1.01576 0.90625 0.73232 0.65554 0.6000 0.51929 0.38996 0.28499 0.24838 0.15882
Proportion of Variance 0.3868 0.1741 0.1165 0.07017 0.06878 0.05475 0.03575 0.02865 0.0240 0.01798 0.01014 0.00541 0.00411 0.00168
Cumulative Proportion  0.3868 0.5609 0.6775 0.74763 0.81641 0.87116 0.90692 0.93556 0.9596 0.97755 0.98768 0.99310 0.99721 0.99889
                          PC15
Standard deviation     0.12885
Proportion of Variance 0.00111
Cumulative Proportion  1.00000
```

Figure 10: componentes

3.4 Grafico de Sedimentacion

El siguiente gráfico de sedimentación nos permite observar visualmente cuantos componentes podemos tomar de tal forma que nos permitan considerar un gran porcentaje de información. (Partiendo desde un 65 o 70 porciento)

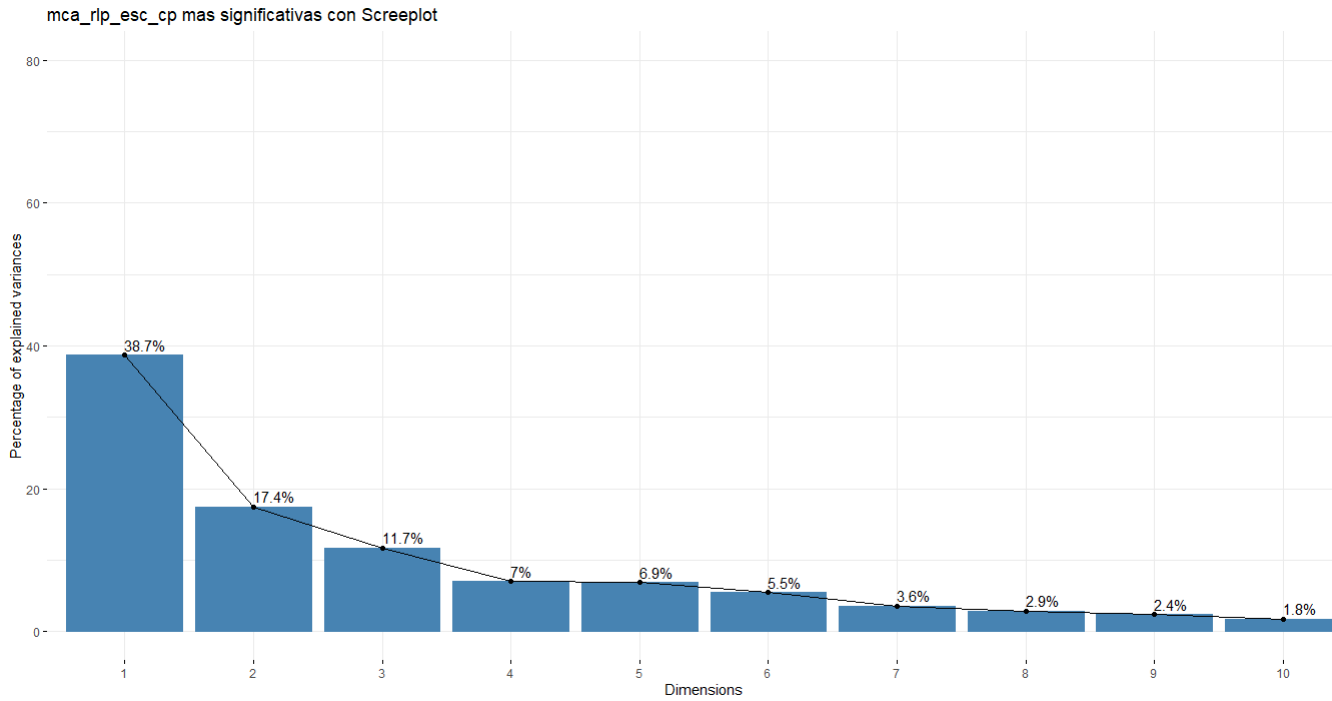

Figure 11: Grafico Barras Sedimentacion

3.5 Biplot

El grafico biplot busca represetnar en un grafico bidimensional las variables explicadas por 2 componentes.En el siguiente caso se observa que las variables fosforo, dbo, turbiedad, cadmio y oxido disuelto estan explicadas por el componente 1.Por otro lado las temperaturas, el nitrato, fosfato y el aire son explicadas por el componente 2.Por otro lado podemos observar que tambien hay variables que no son explicadas por los componentes 1 y 2 sino que son explicados por otros componentes, como por ejemplo, clorofila y enterococos.

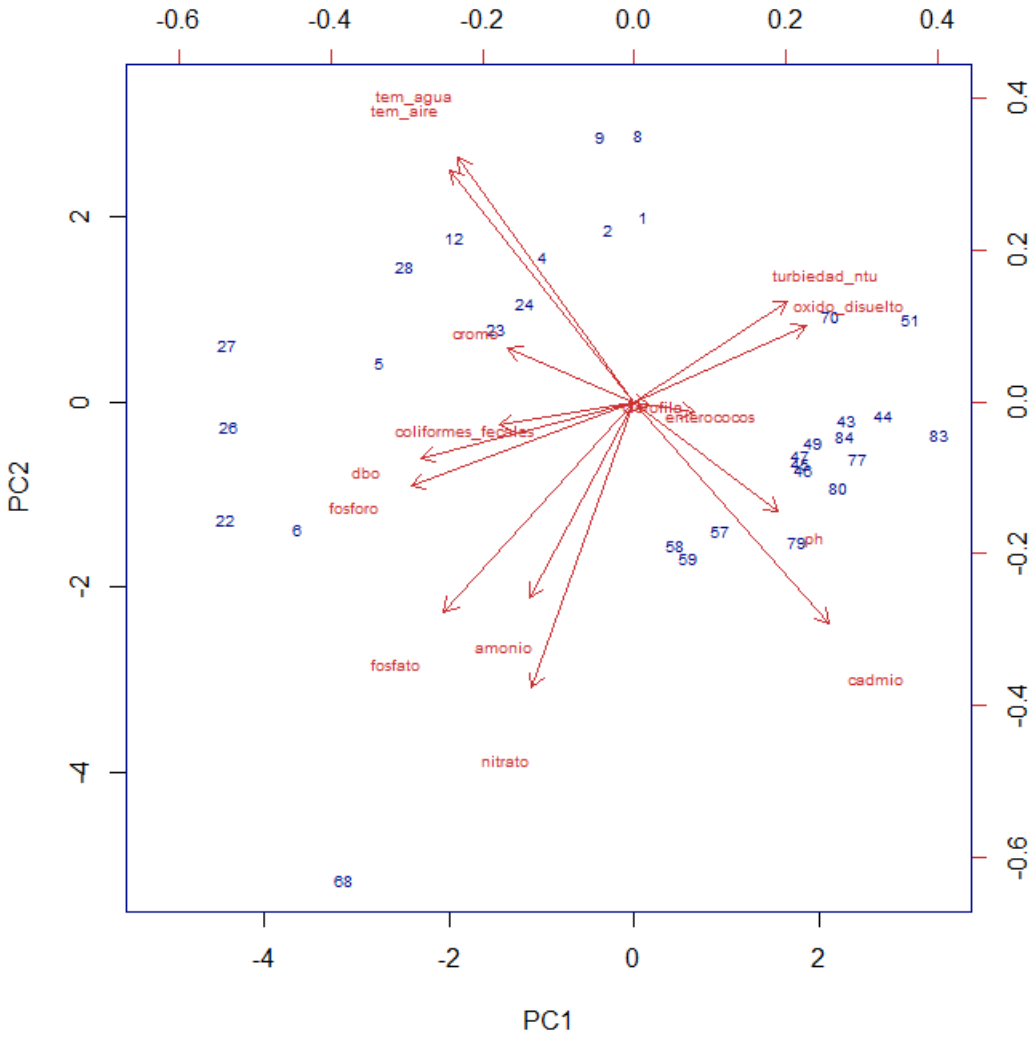

Figure 12: Biplot

Clustering

4.1 Clustering jerárquico

El siguiente dendograma nos indica como se agrupan los datos segun sus similitudes y diferencias. las distancias se pueden ver por la longitud de las ramas lo que se observa similitud aquellas que estan mas juntas

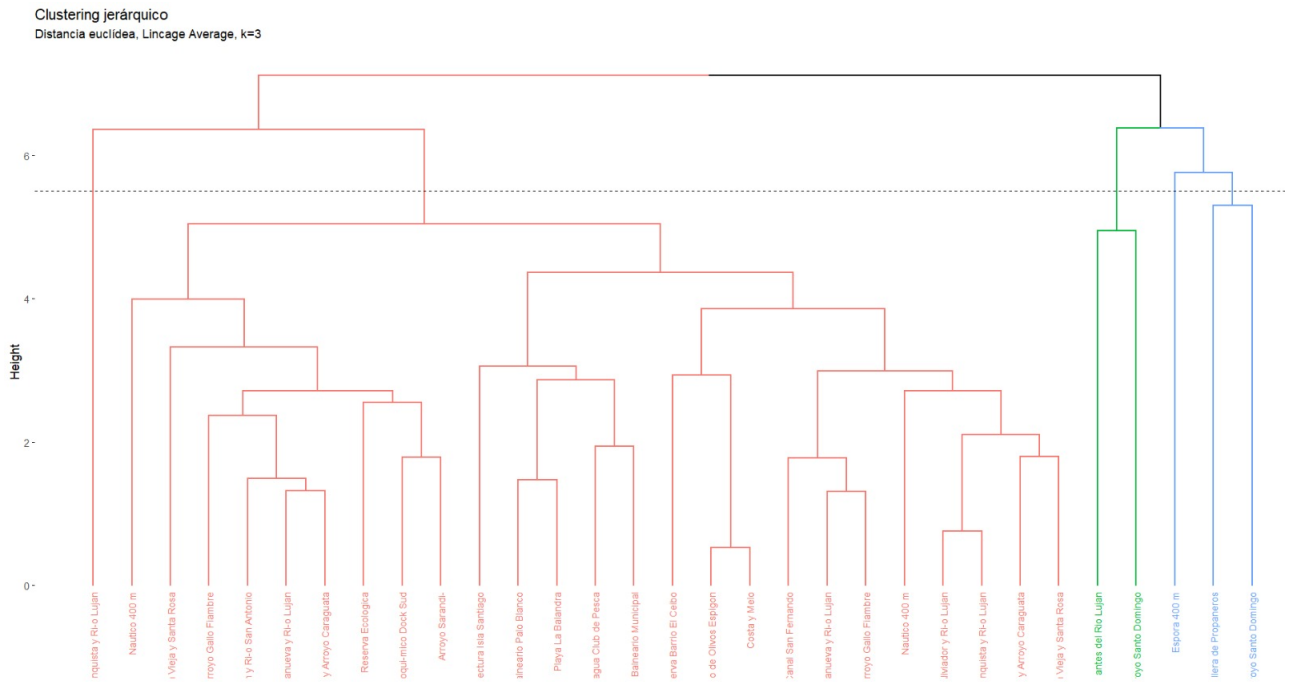

Figure 13: Dendograma Euclidea

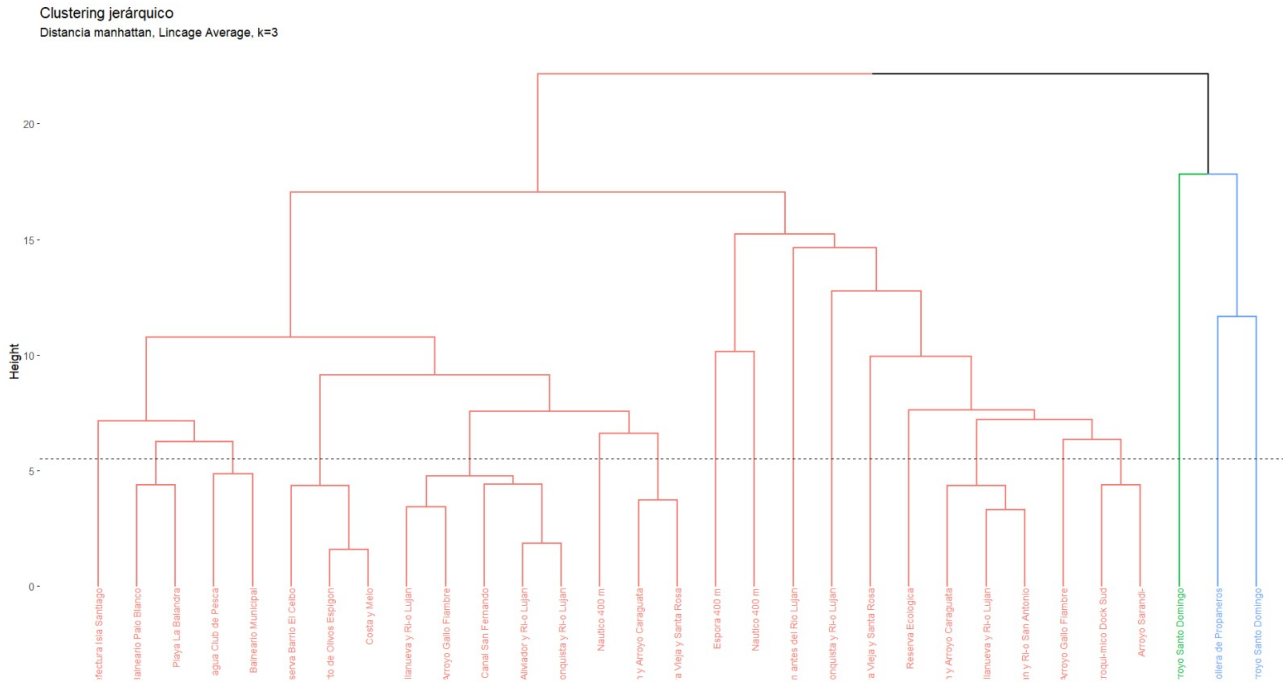

Figure 14: Dendograma Manhatta

4.2 Clustering no jerárquico : K-means

El siguiente grafico indica la forma en que se dividen los datos en funcion de sus caracteristicas utilizando funcion kmeans seleccionando 3 centroides

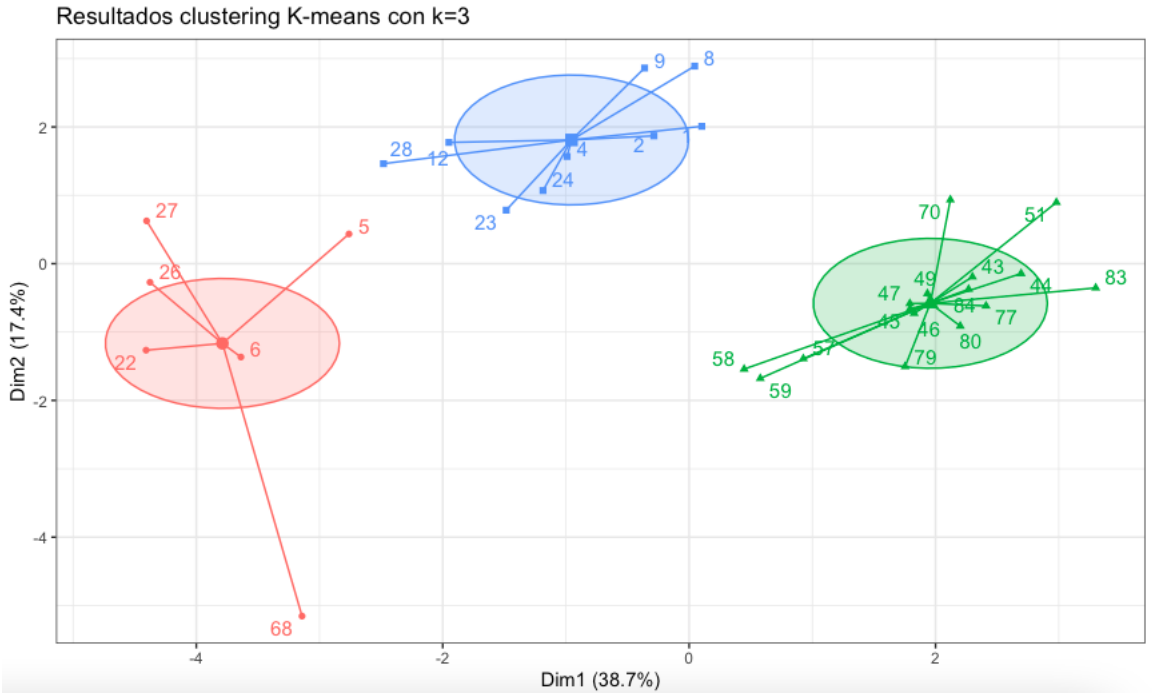

Figure 15: K-means

Conclusión

Para empezar realizamos varios análisis, un summary inicial para observar la distribución de las variables, en donde detectamos outliers y la visualizacion de la correlacion nos permitio ver la relacion entre las variables.
A pesar del reducido número de observaciones se pudo realizar el metodo de clustering jerarquico y k means, ademas de visualizar los cluster graficamente..
A la vez afianzamos nuestro dominio sobre la herramienta R.

En el presente informe queda comprobado la gran importancia de la toma inicial de los datos en los centros de monitoreo. Como se explico durante la presentacion, y visualizaciones mediante, una mala toma en la muestra puede significar informacion errada.

Anexo

```
1
2
3
4  #METADATOS Y DOCUMENTACION
5  #ICA: https://monitorpisa.acumar.gob.ar/sistema-de-indicadores/calidad-ambiental/indice-
       de-calidad-de-agua-superficial-uso-iv/
6  #
       ##############################################################################################

7  ############################### 01 PAQUETES NECESARIOS
       ###################################
8  #
       ##############################################################################################

9  #1.1 - INSTALACION DE PAQUETES
10
11 library(readr)
12 library(dplyr)
13 library(psych)
14 library(tidyverse)
15 library(sqldf)
16 library(xtable)
17 library(factoextra)
18
19 #
       ##############################################################################################

20 ##################### 02. CARGA, TRANSFORMACION Y LIMPIEZA DE DATOS
       ######################
21 #
       ##############################################################################################

22 rm(mca_rlp)
23 mca_rlp <- read.table("./agc_y_riodelaplata2023_2da_camp.csv",sep=";",dec=".",header = T)
24 names(mca_rlp)
25 mca_rlp<-mca_rlp[, c(-1,-2,-3,-4,-5,-6,-30,-31)]
26 names(mca_rlp)
27
28 #colnames(mca_rlp)<-c("Zona","campana","tem_agua","tem_aire","oxi_disu","ph","f_olor","f_
       color",
29 #                            "f_espum","f_mat_flot", "col_fecales","escher_coli_ufc","
       enteroc_ufc",
```

```
#                             "nitrato_mg","amonio","fosforo_tot","fosforo_ort","clorofila","microcistina",
#                             "turbiedad_ntu","hidr_deriv_petro","cromo_total","cadmio_total","clorofila_ug",
#                             "microcistina_ug", "ica_num", "ica_det")

colnames(mca_rlp)<-c("tem_agua","tem_aire","oxido_disuelto","ph","f_olor","f_color",
                     "f_espum","f_mat_flot", "coliformes_fecales","escherichia_coli","enterococos",
                     "nitrato","amonio","fosforo","fosfato","dbo","dqo",
                     "turbiedad_ntu","hidrocarburos_petro","cromo","cadmio","clorofila",
                     "microcistina")

View(mca_rlp)
names(mca_rlp)

attach(mca_rlp)
#mca_rlp$Zona <- stri_replace_all_regex(mca_rlp$Zona,
#                                       pattern=c('     ','    ','    ','=>',
       '    ','    ','    ','    ','    ','  ','=<'),
#                                       replacement=c('E','e','a','mayor a ','ni','',   'o','u', 'er','i','menor a '),
#                                       vectorize=FALSE)

#mca_rlp$campana <- stri_replace_all_regex(mca_rlp$campana,
#                                          pattern=c('     ','    ','    ','=>',
     '    ','    ','    ','    ','    ','  ','=<'),
#                                          replacement=c('E','e','a','mayor a ','ni','',   'o','u', 'er','i','menor a '),
#                                          vectorize=FALSE)

mca_rlp[mca_rlp == "no se midi  "] <- NA
mca_rlp[mca_rlp == "no se muestre  "] <- NA
mca_rlp[mca_rlp == "sin muestra"] <- NA
mca_rlp[mca_rlp == "sin equipo"] <- NA
mca_rlp[mca_rlp == "no funcion  "] <- NA
mca_rlp[mca_rlp == "N/R"] <- NA
mca_rlp[mca_rlp == "falto un frasco"] <- NA
mca_rlp[mca_rlp == ""] <- NA
mca_rlp[mca_rlp == "no se pudo calcular"] <- NA

#
    ----------------------------------------------------------------------------------------------

#ANALISIS PARA CONTROLAR CUANTA INFORMACION SE PERDERIA AL ELIMINAR NAs
#rm(prueba)
#prueba <- mca_rlp
#prueba<-mca_rlp[, c(-24)]
#ELIMINAMOS LA COLUMNA DE ICA PORQUE TIENE DEMASIADOS NAs. SI DEJAMOS LA COLUMNA
#ICA Y ELIMINAMOS TODOS LOS NAs NOS RESULTAN 43OBSERVACIONES (51% DE PERDIDA) LAS RESTANTES.
#SIN EMBARGO, SI ELIMINAMOS ICA, Y LUEGO ELIMINAMOS TODOS LOS NAs, NOS RESULTAN 53 OBSERVACIONES

#View(summarise_all(prueba, funs(sum(is.na(.)))))
```

```
#EN INFORME: SE PIERDE UN 36% DE LOS DATOS DESPUES DE ELIMINAR LOS NA
#
    ------------------------------------------------------------------------------------------

#REEMPLAZAR LOS CARACTERES ESPECIALES POR VACIOS
names(mca_rlp)
mca_rlp$tem_agua <- stri_replace_all_regex(mca_rlp$tem_agua,
                                           pattern=c('=<', "<", ">=", ">"),
                                           replacement=c(''),
                                           vectorize=FALSE)

mca_rlp$tem_aire <- stri_replace_all_regex(mca_rlp$tem_aire,
                                           pattern=c('=<', "<", ">=", ">"),
                                           replacement=c(''),
                                           vectorize=FALSE)

mca_rlp$oxido_disuelto <- stri_replace_all_regex(mca_rlp$oxido_disuelto,
                                           pattern=c('=<', "<", ">=", ">"),
                                           replacement=c(''),
                                           vectorize=FALSE)

mca_rlp$nitrato <- stri_replace_all_regex(mca_rlp$nitrato,
                                           pattern=c('=<', "<", ">=", ">"),
                                           replacement=c(''),
                                           vectorize=FALSE)

mca_rlp$amonio <- stri_replace_all_regex(mca_rlp$amonio,
                                           pattern=c('=<', "<", ">=", ">"),
                                           replacement=c(''),
                                           vectorize=FALSE)

mca_rlp$fosforo <- stri_replace_all_regex(mca_rlp$fosforo,
                                          pattern=c('=<', "<", ">=", ">"),
                                          replacement=c(''),
                                          vectorize=FALSE)


mca_rlp$fosfato <- stri_replace_all_regex(mca_rlp$fosfato,
                                          pattern=c('=<', "<", ">=", ">"),
                                          replacement=c(''),
                                          vectorize=FALSE)

mca_rlp$dbo <- stri_replace_all_regex(mca_rlp$dbo,
                                          pattern=c('=<', "<", ">=", ">"),
                                          replacement=c(''),
                                          vectorize=FALSE)

mca_rlp$dqo <- stri_replace_all_regex(mca_rlp$dqo,
                                           pattern=c('=<', "<", ">=", ">"),
                                           replacement=c(''),
                                           vectorize=FALSE)

mca_rlp$turbiedad_ntu <- stri_replace_all_regex(mca_rlp$turbiedad_ntu,
                                           pattern=c('=<', "<", ">=", ">"),
                                           replacement=c(''),
```

```
                                                    vectorize=FALSE)

mca_rlp$hidrocarburos_petro <- stri_replace_all_regex(mca_rlp$hidrocarburos_petro,
                                                      pattern=c('=<', "<", ">=", ">"),
                                                      replacement=c(''),
                                                      vectorize=FALSE)

mca_rlp$cromo <- stri_replace_all_regex(mca_rlp$cromo,
                                                      pattern=c('=<', "<", ">=", ">"),
                                                      replacement=c(''),
                                                      vectorize=FALSE)

mca_rlp$cadmio <- stri_replace_all_regex(mca_rlp$cadmio,
                                                    pattern=c('=<', "<", ">=", ">"),
                                                    replacement=c(''),
                                                    vectorize=FALSE)

mca_rlp$clorofila <- stri_replace_all_regex(mca_rlp$clorofila,
                                                    pattern=c('=<', "<", ">=", ">"),
                                                    replacement=c(''),
                                                    vectorize=FALSE)

mca_rlp$microcistina <- stri_replace_all_regex(mca_rlp$microcistina,
                                                     pattern=c('=<', "<", ">=", ">"),
                                                     replacement=c(''),
                                                     vectorize=FALSE)

#SE CUANTIFICA LAS VARIABLES CATEGORICAS: VARIABLES DUMMY
mca_rlp[mca_rlp == "Presencia"] <- 1
mca_rlp[mca_rlp == "Ausencia"] <- 0
#
    ----------------------------------------------------------------------------------------

#CASTEO DE DATOS:
str(mca_rlp)
names(mca_rlp)

#Casteo de datos
mca_rlp$tem_agua <- suppressWarnings(as.numeric(mca_rlp$tem_agua))
mca_rlp$tem_aire <- suppressWarnings(as.numeric(mca_rlp$tem_aire))
mca_rlp$oxido_disuelto <- suppressWarnings(as.numeric(mca_rlp$oxido_disuelto))
mca_rlp$ph <- suppressWarnings(as.numeric(mca_rlp$ph))
mca_rlp$f_olor <- suppressWarnings(as.numeric(mca_rlp$f_olor))
mca_rlp$f_color <- suppressWarnings(as.numeric(mca_rlp$f_color))
mca_rlp$f_espum <- suppressWarnings(as.numeric(mca_rlp$f_espum))
mca_rlp$f_mat_flot <- suppressWarnings(as.numeric(mca_rlp$f_mat_flot))
mca_rlp$coliformes_fecales <- suppressWarnings(as.numeric(mca_rlp$coliformes_fecales))
mca_rlp$escherichia_coli <- suppressWarnings(as.numeric(mca_rlp$escherichia_coli))
mca_rlp$enterococos <- suppressWarnings(as.numeric(mca_rlp$enterococos))
mca_rlp$nitrato <- suppressWarnings(as.numeric(mca_rlp$nitrato))
mca_rlp$amonio <- suppressWarnings(as.numeric(mca_rlp$amonio))
mca_rlp$fosforo <- suppressWarnings(as.numeric(mca_rlp$fosforo))
mca_rlp$fosfato <- suppressWarnings(as.numeric(mca_rlp$fosfato))
mca_rlp$dbo <- suppressWarnings(as.numeric(mca_rlp$dbo))
mca_rlp$dqo <- suppressWarnings(as.numeric(mca_rlp$dqo))
```

```r
mca_rlp$turbiedad_ntu <- suppressWarnings(as.numeric(mca_rlp$turbiedad_ntu))
mca_rlp$hidrocarburos_petro <- suppressWarnings(as.numeric(mca_rlp$hidrocarburos_petro))
mca_rlp$cromo <- suppressWarnings(as.numeric(mca_rlp$cromo))
mca_rlp$cadmio <- suppressWarnings(as.numeric(mca_rlp$cadmio))
mca_rlp$clorofila <- suppressWarnings(as.numeric(mca_rlp$clorofila))
mca_rlp$microcistina <- suppressWarnings(as.numeric(mca_rlp$microcistina))

str(mca_rlp)
summary(mca_rlp)
names(mca_rlp)
View(summarise_all(mca_rlp, funs(sum(is.na(.)))))
mca_rlp <- na.omit(mca_rlp)

#BORRO LOS FLAGS
mca_rlp<-mca_rlp[, c(-5,-6,-7,-8)]

#QUE DEBEMOS CONSIDERAR PARA ELIMINAR UNA COLUMNA?:
#MUCHAS NAs; OUTLIERS; VALOR CONSTANTE
#AJUSTE: NO SIGNIFICANTE EN LA MATRIZ DE CORR
#
    ----------------------------------------------------------------------------------------

#BOXPLOT DE LOS DATOS CUANTITATIVOS:
#CON GRAFICO DINAMICO:
summary(mca_rlp)
names(mca_rlp)
rm(mca_rlp_boxplot)
mca_rlp_boxplot <- plot_ly(y = ~mca_rlp$microcistina, type = "box")
mca_rlp_boxplot

#col_fecales; microcistina; escher_coli_ufc; enteroc_ufc; nitrato_mg; amonio; turbiedad_
    ntu; clorofila_ug

#SUMMARY DE OUTLIERS:
#rm(mca_rlp_out)
#names(mca_rlp_out)
#mca_rlp_out <- mca_rlp[,c(10,11,13,17,18,22)]
#summary(mca_rlp_out)
#xtable(summary(mca_rlp_out) ) #  para sacar en latex

par(mfrow=c(3, 3))
mca_rlp_escher_coliformes_fecales_out <- boxplot(mca_rlp$coliformes_fecales, col="skyblue
    ", frame.plot=F)
mca_rlp_escher_escherichia_coli_out <- boxplot(mca_rlp$escherichia_coli, col="skyblue",
    frame.plot=F)
mca_rlp_enterococos_out <- boxplot(mca_rlp$enterococos, col="skyblue", frame.plot=F)
mca_rlp_nitrato_out <- boxplot(mca_rlp$nitrato, col="skyblue", frame.plot=F)
mca_rlp_amonio_out <- boxplot(mca_rlp$amonio, col="skyblue", frame.plot=F)
mca_rlp_dqo_out <- boxplot(mca_rlp$dqo, col="skyblue", frame.plot=F)
mca_rlp_turbiedad_ntu_out <- boxplot(mca_rlp$turbiedad_ntu, col="skyblue", frame.plot=F)
mca_rlp_clorofila_out <- boxplot(mca_rlp$clorofila, col="skyblue", frame.plot=F)
mca_rlp_microcistina_out <- boxplot(mca_rlp$microcistina, col="skyblue", frame.plot=F)

mca_rlp_microcistina_out$out

```

```r
#ELIMINO LOS OUTLIERS
mca_rlp <- mca_rlp[!(mca_rlp$coliformes_fecales %in% c(32000,22000,21000)),] #PROBAR ASI, SINO BORRAR LAS DE 15K
mca_rlp <- mca_rlp[!(mca_rlp$escherichia_coli %in% c(32000,11200,7200,6800)),] #PROBAR ASI, SINO BORRAR 5600
mca_rlp <- mca_rlp[!(mca_rlp$enterococos %in% c(9300,5000,5500)),]
mca_rlp <- mca_rlp[!(mca_rlp$nitrato %in% c(44.0)),]
mca_rlp <- mca_rlp[!(mca_rlp$amonio %in% c(18.0,11.0,6.3)),]
mca_rlp <- mca_rlp[!(mca_rlp$dqo %in% c(230,90,70)),] #** PROBAR, SINO BORRAR TODA LA COLUMNA O LOS 59 PARA ABAJO
mca_rlp <- mca_rlp[!(mca_rlp$turbiedad_ntu %in% c(432)),]
mca_rlp <- mca_rlp[!(mca_rlp$clorofila %in% c(740.93,152.11)),]
mca_rlp <- mca_rlp[!(mca_rlp$microcistina %in% c(5.00,2.98,2.61,0.99,0.90)),]#**PROBAR ASI, SINO BORRAR LA COLUMNA

#PIERDO 11 REGISTROS: de 53 pasamos a 42

#CONTROL:
par(mfrow=c(3, 3))
mca_rlp_escher_coliformes_fecales_out <- boxplot(mca_rlp$coliformes_fecales, col="skyblue", frame.plot=F)
mca_rlp_escher_escherichia_coli_out <- boxplot(mca_rlp$escherichia_coli, col="skyblue", frame.plot=F)
mca_rlp_enterococos_out <- boxplot(mca_rlp$enterococos, col="skyblue", frame.plot=F)
mca_rlp_nitrato_out <- boxplot(mca_rlp$nitrato, col="skyblue", frame.plot=F)
mca_rlp_amonio_out <- boxplot(mca_rlp$amonio, col="skyblue", frame.plot=F)
mca_rlp_dqo_out <- boxplot(mca_rlp$dqo, col="skyblue", frame.plot=F)
mca_rlp_turbiedad_ntu_out <- boxplot(mca_rlp$turbiedad_ntu, col="skyblue", frame.plot=F)
mca_rlp_clorofila_out <- boxplot(mca_rlp$clorofila, col="skyblue", frame.plot=F)
mca_rlp_microcistina_out <- boxplot(mca_rlp$microcistina, col="skyblue", frame.plot=F)

rm(mca_rlp_escher_coliformes_fecales_out,mca_rlp_escher_escherichia_coli_out,
   mca_rlp_enterococos_out,mca_rlp_nitrato_out,
   mca_rlp_amonio_out,mca_rlp_dqo_out,
   mca_rlp_turbiedad_ntu_out,mca_rlp_clorofila_out,
   mca_rlp_microcistina_out)

#ELIMINO LOS NAs
mca_rlp <- na.omit(mca_rlp)
#
    ##############################################################################################

#################### 03. ANALISIS DE COMPONENTES PRINCIPALES
    ############################
#
    ##############################################################################################

set.seed(101)

#3.1 - ESCALAMOS LOS DATOS:
names(mca_rlp)

#3.2.3 - AJUSTAMOS ELIMINANDO LOS FLAGS DE PRESENCIA DE OLOR, COLOR, ESPUMA Y METALES FLOTANTES
#3.1.2 - ELIMINAMOS hidr_deriv_petro PORQUE MANEJA UN INDICE MUY BAJO QUE NO PERMITE
```

```
    ESCALAR
rm(mca_rlp_ajus)
mca_rlp_ajus <- mca_rlp[, c(-6,-13,-15,-19)]

rm(mca_rlp_esc)
mca_rlp_esc <- scale(mca_rlp_ajus)
#
    ----------------------------------------------------------------------------------------

#3.2 - MATRIZ DE CORRELACION:
#Correlaciones para justificar el Aepf_cp (argumentar)
#cor(epf[, -1]) # matriz de correlacion
#corrplot(cor(epf[, -1])) # scatter plot de correlaciones
rm(mca_rlp_esc_cor)
mca_rlp_esc_cor <- cor(mca_rlp_esc) #requiere corrplot. Las variables deben ser
    num ricas.
mca_rlp_esc_cor

#Visualizacion con indice de correlacion para cada atributo
corrplot(mca_rlp_esc_cor, method="number",tl.col="black",tl.cex=0.7 )

#VERIFICAR hidrocarburos_petro; CROMO; CADMIO ***************
#
    ----------------------------------------------------------------------------------------

#Es relevante aplicar Aepf_cp?: Se comprueba mediante un test de Barlett
#N = Cantidad registros
#
# La prueba de esfericidad de Bartlett prueba
#H0: no hay correlaciones (esfericidad) por lo que si pvalor chico entonces est
    habilitado Aepf_cp
cortest.bartlett(cor(mca_rlp_esc),n=31) #n=tama o de muestra
#-------------------------------------------------------------------------------
#KMO  #Kaiser-Meyer-Olkin analiza los autovalores de la matriz de covarianzas
#sirve para comparar los valores de correlacion de las variables y sus correlaciones
    parciales
#si es cercano a 1, tiene sentido el analisis de componentes principales.
KMO(cor(mca_rlp_esc))

#-------------------------------------------------------------------------------
### Aepf_cp usando Rbase
# La funci n prcomp() calcula autom ticamente el valor de las
# componentes principales para cada observaci n
rm(mca_rlp_esc_cp)
mca_rlp_esc_cp <- prcomp(mca_rlp_esc, scale = FALSE) # Analizo los componentes
    principales.
# Por defecto, prcomp() centra las variables para que tengan media cero
# si se quiere adem s que su desviaci n est ndar sea de uno, hay que indicar scale =
    TRUE.
summary(mca_rlp_esc_cp) #Obtenemos el porcentaje de explicacion de los Aepf_cp *****
names(mca_rlp_esc_cp)
# Los elementos center y scale almacenados en el objeto pca contienen la media y
    desviaci n t pica
# de las variables previa estandarizaci n (en la escala original).
mca_rlp_esc_cp$center
```

```
mca_rlp_esc_cp$scale
mca_rlp_esc_cp$sdev
# rotation contiene el valor de los autovalores para cada componente (eigenvector).
# El n mero m ximo de componentes principales se corresponde con el m nimo(n-1,p),
# que en este caso es min(24,9)= 9.
mca_rlp_esc_cp$rotation  # **********
mca_rlp_esc_cp$x #autovectores ***************

#-------------------------------------------------------------------------------
# Grafico de Sedimentacion de las componentes
plot(mca_rlp_esc_cp,
     type="l",
     main="Gr fico de sedimentaci n",
     col=c("blue4"))
abline(0.7,0,col=c("brown3")) # linea horizontal en 1 del eje y.

#4.6.2 - Usamos un grafico de barras. USAR ESTE EN INFORME
fviz_screeplot(mca_rlp_esc_cp,
               addlabels = TRUE,
               ylim = c(0, 80),
               main="mca_rlp_esc_cp mas significativas con Screeplot")

#para graficar autovalores ordenados (gr fico de sedimentaci n)
#fviz_screeplot(epf_cp, addlabels = TRUE, ylim = c(0, 60))
#
    ------------------------------------------------------------------------------------------

#4.7 - GRAFICO DE BIPLOT
biplot(x = mca_rlp_esc_cp, scale = 0, cex = 0.6, col = c("blue4", "brown3"))

# Biplot con puntos. Se ven las variables y los casos.
# El grafico entre la Componente Principal 1 y 2, se puede apreciar
# dos grandes agrupamientos de variables, indicando correlacion positiva en
# cada grupo, y que estos grupos estan de forma perpendicular, indicando correlacion nula
#biplot(x = mca_rlp_esc_cp, scale = 0, cex = 0.6, xlabs=rep(".", nrow(mca_rlp_esc)),col =
     c("grey", "brown3"))
#
    ##############################################################################################

################################### APLICACION DE CLUSTERING
    ################################
#
    ##############################################################################################

names(mca_rlp_ajus_esc)
str(mca_rlp_ajus)
View(mca_rlp_esc)
summary(mca_rlp_esc)
#
    ##############################################################################################

################################ 01 CLUSTERING JERARQUICO
    ####################################
#
    ##############################################################################################
```

```r
#1.1 - DEFINIMOS EL TIPO DE DISTANCIA: PROBAR CON OTROS TIPOS DE DISTANCIAS:
rm(mca_rlp_esc_jer_dist_eu)
mca_rlp_esc_jer_dist_eu <- dist(x = mca_rlp_esc, method = "euclidean")
mca_rlp_esc_jer_dist_man <- dist(x = mca_rlp_esc, method = "manhattan")
mca_rlp_esc_jer_dist_eu
#
    ----------------------------------------------------------------------------------------

#2.1 - DEFINIMOS EL LINKAJE COMPLETO O POR PROMEDIO: PROBAR OTROS LINCAJES (WARD|
    CENTROIDE|...)
rm(agc_ajus_esc_jer_dist_eu_completo)
mca_rlp_esc_jer_dist_eu_completo <- hclust(d = mca_rlp_esc_jer_dist_eu, method = "
    complete")
mca_rlp_esc_jer_dist_eu_average  <- hclust(d = mca_rlp_esc_jer_dist_eu, method = "average
    ")
mca_rlp_esc_jer_dist_eu_completo
mca_rlp_esc_jer_dist_eu_average

summary(mca_rlp_ajus_esc)

mca_rlp_esc_jer_dist_man_completo <- hclust(d = mca_rlp_esc_jer_dist_man, method = "
    complete")
mca_rlp_esc_jer_dist_man_average  <- hclust(d = mca_rlp_esc_jer_dist_man, method = "
    average")
#
    ----------------------------------------------------------------------------------------

#3.1 - EL COEFICIENTE COFRENETICO MIRA LAS CORRELACIONES: Se quiere que sea cercana a 1
# X = AL METODO DE DISTANCIA; COFRENETICO = AL METODO DE LINKAJE
cor(x = mca_rlp_esc_jer_dist_eu, cophenetic(mca_rlp_esc_jer_dist_eu_completo))
cor(x = mca_rlp_esc_jer_dist_eu, cophenetic(mca_rlp_esc_jer_dist_eu_average))

cor(x = mca_rlp_esc_jer_dist_man, cophenetic(mca_rlp_esc_jer_dist_man_completo))
cor(x = mca_rlp_esc_jer_dist_man, cophenetic(mca_rlp_esc_jer_dist_man_average))
#CONCLUSION: ME QUEDO CON EUCLIDEA AVERAGE PORQUE ME DA MEJOR EL COEFICIENTE COFRENETICO
#
    ----------------------------------------------------------------------------------------

#04 Graficamos el dendrograma con distancia euclidea y linkaje average
fviz_dend(x = mca_rlp_esc_jer_dist_eu_average, k = 3, cex = 0.6) +
  geom_hline(yintercept = 5.5, linetype = "dashed") +
  labs(title = "Clustering jer rquico",
       subtitle = "Distancia eucl dea , Lincage Average, k=3")

fviz_dend(x = mca_rlp_esc_jer_dist_man_average, k = 3, cex = 0.6) +
  geom_hline(yintercept = 5.5, linetype = "dashed") +
  labs(title = "Clustering jer rquico",
       subtitle = "Distancia manhattan, Lincage Average, k=3")

#para ver a q u  grupo se asign  cada caso (CASO = observacion por cluster):
cutree(mca_rlp_ajus_esc_jer_dist_eucli_average, k = 3)#[5] #ENTRE CORCHETES MIRAMOS LA
    POS. DE LA OBS

#
```

```
    ------------------------------------------------------------------------------------------

#
    ------------------------------------------------------------------------------------------

#
    ------------------------------------------------------------------------------------------

################################################################################
############################## CLUSTER POR K-MEANS ##############################
################################################################################
library(broom)
library(factoextra)

set.seed(101)
#-------------------------------------------------------------------------------
km_cluster_mca_rlp_eu <- kmeans(x = mca_rlp_esc_jer_dist_eu, centers = 3, nstart = 50)
km_cluster_mca_rlp_eu

km_cluster_mca_rlp_man <- kmeans(x = mca_rlp_esc_jer_dist_man, centers = 3, nstart = 50)
km_cluster_mca_rlp_man
#-------------------------------------------------------------------------------
#???
fviz_nbclust(x = mca_rlp_esc , FUNcluster = kmeans, method = "silhouette", k.max = 11) +
  labs(title = "Numero optimo de clusters", diss = mca_rlp_esc_jer_dist_eu)

fviz_cluster(object = km_cluster_mca_rlp_eu, data = mca_rlp_esc, show.clust.cent = TRUE,
             ellipse.type = "euclid", star.plot = TRUE, repel = TRUE) +
  labs(title = "Resultados clustering K-means con k=3 y distancia eucledean") +
  theme_bw() +
  theme(legend.position = "none")

fviz_cluster(object = km_cluster_mca_rlp_man, data = mca_rlp_esc, show.clust.cent = TRUE,
             ellipse.type = "manhattan", star.plot = TRUE, repel = TRUE) +
  labs(title = "Resultados clustering K-means con k=3 y distancia manhattan") +
  theme_bw() +
  theme(legend.position = "none")

#-------------------------------------------------------------------------------
#library(NbClust)

#km_clusters <- eclust(x = mca_rlp_esc, FUNcluster = "kmeans", k = 2, # Resultados para K
     = 2, seed = 123,
#                      hc_metric = "manhattan", nstart = 50, graph = FALSE)

#fviz_silhouette(sil.obj = km_clusters, print.summary = TRUE, palette = "jco",
#                ggtheme = theme_classic())
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
#-------------------------------------------------------------------------------
```