



# **Instituto Politécnico Nacional Escuela Superior de Computo**

**Programación para la ciencia de datos.**

**Cristal Karina Galindo Durán**

**Practica 10:**

**Análisis de Conglomerados**

**Vianey Maravilla Pérez**

**3AM1**

**Unidad temática a la que corresponde la práctica.** III. Técnicas y métodos de modelado.

**Objetivo.** Realizar scripts en Lenguaje R que permita aplicar análisis de conglomerados jerárquicos y no jerárquicos sobre un conjunto de datos

**Introducción.**

El análisis de conglomerados o también llamado análisis de clúster es una técnica multivariante que permite formar grupos; asimismo, se complementa con técnicas que permiten la visualización de los grupos conformados.

El análisis de clúster cuenta con algoritmos de agrupamiento divididos en: jerárquicos y no jerárquicos.

Los algoritmos de agrupamiento jerárquicos cuentan con grupos anidados y la agrupación final tienen un conjunto de grupos crecientes.

Por su parte, los algoritmos de agrupamiento no jerárquicos comienzan con una solución inicial, un número de grupos  $g$  fijado y agrupa los objetos para obtener los  $g$  grupos.

En esta práctica se proponen una serie de ejercicios con la finalidad de que el discente ponga en práctica los conocimientos adquiridos sobre el tema en cuestión.

**Material o equipo necesario**

- Computadora
- Internet
- Lenguaje R y R Studio

**Consideraciones:**

Para lograr tener una eficiencia buena al realizar la práctica, lo primero que debemos considerar es que el Análisis de Clúster es un análisis multivariante Independiente que nos permitirá formar grupos y clasificarlos, para ello debemos de seleccionar las medidas de distancia o similitud que mejor le convengan a nuestros datos.

En este caso podemos implementar la medida Euclidiana puesto que es la distancia en línea recta o la trayectoria más corta posible entre dos puntos

## Ejercicios

1.-

a) Considera los datos del siguiente enlace:

<https://archive.ics.uci.edu/ml/machine-learning-databases/00445/>

b) Realiza un análisis exploratorio de los datos

c) Elige la medida de distancia recomendada de acuerdo al tipo de datos

d) Aplica el algoritmo jerárquico

e) Aplica el algoritmo K-Means

f) Aplica el algoritmo del codo para saber el número adecuado de grupos

g) Obtén la relación de los individuos que pertenecen a cada grupo

h) Realiza una comparativa entre los algoritmos

## Procedimiento:

```
1 library(readr)
2 library(tidyverse) # Manipulación de datos
3 library(factoextra) # Librería de clusterización y visualización
4 library(openxlsx) # Librería que interactúa con MS Excel
5 library(corrplot) # Librería para el gráfico de correlaciones
6 library(corr) # Otra opción de librería para el cálculo y gráfico de correlaciones
7 library(psych)
8 library(ggcorrplot)
9 library(stats) # Librería del sistema base
10 library(polycor)
11 library(GPArotation)
12 library(cluster) # Algoritmos de clusterización
13 library(gridExtra)
14 library(ggplot2)
15
16 principal <- function()
17 {
18
19   # Cargar los datos del archivo
20   #read_csv("C:/Users/viane/Desktop/ESCOM/3.-TERCER SEMESTRE/PROGRAMACION PARA LAS CIENCIAS DE DATOS/Absenteeism_at_work.csv")
21   Absenteeism <- read.xlsx(xlsxFile='C:/Users/viane/Desktop/ESCOM/3.-TERCER SEMESTRE/PROGRAMACION PARA LAS CIENCIAS DE DATOS/Absenteeism_at_work.xlsx',
22     sheet = 'Absenteeism_at_work')
23   View(Absenteeism)
24   Absenteeism <- Absenteeism[, -1]
25
26   # Explorar los datos
27   # Desplegar la estructura de los datos
28   str(Absenteeism)
29
30
31   # Obtención de medidas estadísticas
32   summary(Absenteeism)
33
34   # Valores NA
35   any(is.na(Absenteeism))
36 }
```

```

37 # Convertir los datos a un DataFrame
38 Absen<- as.data.frame(scale(Absenteeism))
39
40 # Obtención de medidas estadísticas del DF
41 summary(Absen)
42
43 # Se aplica la medida de distancia para obtener la matriz distancia
44 dist_mat <- dist(Absen, method = 'euclidean')
45
46 # Extra: aplicamos Hierarchical Clustering
47 eucli <- hclust(d = dist(x = Absen, method = "euclidean"),method = "complete")
48 fviz_dend(eucli, k = 3, cex = 0.6) + geom_hline(yintercept = 5.5, linetype = "dashed") +
49 labs(title = "Hierarchical clustering", subtitle = "Distancia euclídea, Lincage complete, K=3")
50
51
52
53
54
55
56
57 # Obtención de los grupos
58 grupos <- hclust(dist_mat, method = "ward.D")
59 plot(grupos)
60
61 # Trazar la línea de corte y mostrar rectángulo en los grupos
62 lineaCorte <- cutree(grupos, k=3)
63
64 plot(grupos)
65 rect.hclust(grupos, k = 3, border = 2:6)
66 abline(h = 3, col = 'red')
67
68 ▾ ##### Se aplica algoritmo K-Means #####
69 grupoK2 <- kmeans(dist_mat, centers = 2, nstart = 25)
70
71 # Obtención de la estructura de los datos K2
72 str(grupoK2)
73
74
75 # Impresión de los grupos
76 print(grupoK2)
77
78 # Obtención de gráfico de los grupos
79 fviz_cluster(grupoK2, data = dist_mat)
80
81 # Resultados de algoritmo K-means
82 set.seed(1234)
83 fviz_cluster(grupoK2, data = dist_mat, show.clust.cent = TRUE, ellipse.type = "euclid", star.plot = TRUE,
84               repel = TRUE) + labs(title = "Resultados clustering K-means") + theme_bw() + theme(legend.position = "none")
85
86 ▾ ##### Método del codo #####
87 set.seed(1234)
88 wcss <- vector()
89 for ( i in 1:20)
90 {
91   wcss[i] <- sum(kmeans(Absen,i)$withinss)
92 }
93
94 ggplot() + geom_point(aes(x = 1:20, y = wcss), color = 'red') + geom_line(aes(x = 1:20, y = wcss), color = 'red') +
95 ggtitle ("Método Del Codo ") + xlab('Cantidad de Centroides K') + ylab('WCSS')
96
97
98 ▾ }
99

```

## RESULTADOS.

```
> library(readr)
> library(tidyverse)# Manipulacion de datos
> library(factoextra)# Librería de clusterización y visualización
> library(openxlsx) # Librería que interactúa con MSExcel
> library(corrplot) # Librería para el gráfico de correlaciones
> library(corr) # Otra opción de librería para el cálculo y gráfico de correlaciones
> library(psych)
> library(ggcorrplot)
> library(stats) # Librería del sistema base
> library(polycor)
> library(GPArotation)
> library(cluster)# Algoritmos de clusterización
> library(gridExtra)
> library(ggplot2)
> # Cargar los datos del archivo
> #read_csv("C:/Users/viane/Desktop/ESCOM/3.-TERCER SEMESTRE/PROGRAMACION PARA LAS CIENCIAS DE DATOS/Absenteeism_at_work.csv")
> Absenteeism <- read.xlsx (xlsxFile='C:/Users/viane/Desktop/ESCOM/3.-TERCER SEMESTRE/PROGRAMACION PARA LAS CIENCIAS DE DATOS/Absenteeism_at_work.xlsx',
+                           sheet = 'Absenteeism_at_work')
> View(Absenteeism)
>
```

^	ID ^	Reason.for.absence ^	Month.of.absence ^	Day.of.the.week ^	Seasons ^	Transportation.expense ^	Distance.from.Residence.to.Work ^	Service.time ^	Age ^	Workload.Average/day ^	Hit.target ^	Disci
1	11	26	7	3	1	289	36	13	33	239554	97	
2	36	0	7	3	1	118	13	18	50	239554	97	
3	3	23	7	4	1	179	51	18	38	239554	97	
4	7	7	7	5	1	279	5	14	39	239554	97	
5	11	23	7	5	1	289	36	13	33	239554	97	
6	3	23	7	6	1	179	51	18	38	239554	97	
7	10	22	7	6	1	361	52	3	28	239554	97	
8	20	23	7	6	1	260	50	11	36	239554	97	
9	14	19	7	2	1	155	12	14	34	239554	97	
10	1	22	7	2	1	235	11	14	37	239554	97	
11	20	1	7	2	1	260	50	11	36	239554	97	
12	20	1	7	3	1	260	50	11	36	239554	97	
13	20	11	7	4	1	260	50	11	36	239554	97	
14	3	11	7	4	1	179	51	18	38	239554	97	
15	3	23	7	4	1	179	51	18	38	239554	97	
16	24	14	7	6	1	246	25	16	41	239554	97	
17	3	23	7	6	1	179	51	18	38	239554	97	
18	3	21	7	2	1	179	51	18	38	239554	97	
19	6	11	7	5	1	189	29	13	33	239554	97	
20	33	23	8	4	1	248	25	14	47	205917	92	
21	18	10	8	4	1	330	16	4	28	205917	92	

```
> Absenteeism <- Absenteeism[, -1]
> # Explorar los datos
> # Desplegar la estructura de los datos
> str(Absenteeism)
'data.frame': 740 obs. of 20 variables:
 $ Reason.for.absence      : num 26 0 23 7 23 23 22 23 19 22 ...
 $ Month.of.absence        : num 7 7 7 7 7 7 7 7 7 7 ...
 $ Day.of.the.week         : num 3 3 4 5 5 6 6 2 2 ...
 $ Seasons                 : num 1 1 1 1 1 1 1 1 1 1 ...
 $ Transportation.expense   : num 289 118 179 279 289 179 361 260 155 235 ...
 $ Distance.from.Residence.to.Work: num 36 13 51 5 36 51 52 50 12 11 ...
 $ Service.time            : num 13 18 18 14 13 18 3 11 14 14 ...
 $ Age                     : num 33 50 38 39 33 38 28 36 34 37 ...
 $ Workload.Average/day    : num 239554 239554 239554 239554 239554 ...
 $ Hit.target              : num 97 97 97 97 97 97 97 97 97 97 ...
 $ Disciplinary.failure     : num 0 1 0 0 0 0 0 0 0 0 ...
 $ Education               : num 1 1 1 1 1 1 1 1 3 ...
 $ Son                     : num 2 1 0 2 2 0 1 4 2 1 ...
 $ Social.drinker          : num 1 1 1 1 1 1 1 1 1 0 ...
 $ Social.smoker           : num 0 0 0 1 0 0 0 0 0 0 ...
 $ Pet                     : num 1 0 0 0 1 0 4 0 0 1 ...
 $ Weight                  : num 90 98 89 68 90 89 80 65 95 88 ...
 $ Height                  : num 172 178 170 168 172 170 172 168 196 172 ...
 $ Body.mass.index         : num 30 31 31 24 30 31 27 23 25 29 ...
 $ Absenteeism.time.in.hours : num 4 0 2 4 2 2 8 4 40 8 ...
```

```

> # Obtención de medidas estadísticas
> summary(Absenteeism)
Reason.for.absence Month.of.absence Day.of.the.week Seasons Transportation.expense Distance.from.Residence.to.Work
Min. : 0.00 Min. : 0.000 Min. :2.000 Min. :1.000 Min. :118.0 Min. : 5.00
1st Qu.:13.00 1st Qu.: 3.000 1st Qu.:3.000 1st Qu.:2.000 1st Qu.:179.0 1st Qu.:16.00
Median :23.00 Median : 6.000 Median :4.000 Median :3.000 Median :225.0 Median :26.00
Mean :19.22 Mean : 6.324 Mean :3.915 Mean :2.545 Mean :221.3 Mean :29.63
3rd Qu.:26.00 3rd Qu.: 9.000 3rd Qu.:5.000 3rd Qu.:4.000 3rd Qu.:260.0 3rd Qu.:50.00
Max. :28.00 Max. :12.000 Max. :6.000 Max. :4.000 Max. :388.0 Max. :52.00

Service.time Age Work.load.Average/day Hit.target Disciplinary.failure Education Son
Min. : 1.00 Min. :27.00 Min. :205917 Min. : 81.00 Min. :0.00000 Min. :1.000 Min. :0.000
1st Qu.: 9.00 1st Qu.:31.00 1st Qu.:244387 1st Qu.: 93.00 1st Qu.:0.00000 1st Qu.:1.000 1st Qu.:0.000
Median :13.00 Median :37.00 Median :264249 Median : 95.00 Median :0.00000 Median :1.000 Median :1.000
Mean :12.55 Mean :36.45 Mean :271490 Mean : 94.59 Mean :0.05405 Mean :1.292 Mean :1.019
3rd Qu.:16.00 3rd Qu.:40.00 3rd Qu.:294217 3rd Qu.: 97.00 3rd Qu.:0.00000 3rd Qu.:1.000 3rd Qu.:2.000
Max. :29.00 Max. :58.00 Max. :378884 Max. :100.00 Max. :1.00000 Max. :4.000 Max. :4.000

Social.drinker Social.smoker Pet Weight Height Body.mass.index Absenteeism.time.in.hours
Min. :0.0000 Min. :0.00000 Min. :0.00000 Min. : 56.00 Min. :163.0 Min. :19.00 Min. : 0.000
1st Qu.:0.0000 1st Qu.:0.00000 1st Qu.:0.00000 1st Qu.: 69.00 1st Qu.:169.0 1st Qu.:24.00 1st Qu.: 2.000
Median :1.0000 Median :0.00000 Median :0.00000 Median : 83.00 Median :170.0 Median :25.00 Median : 3.000
Mean :0.5676 Mean :0.07297 Mean :0.7459 Mean : 79.04 Mean :172.1 Mean :26.68 Mean : 6.924
3rd Qu.:1.0000 3rd Qu.:0.00000 3rd Qu.:1.00000 3rd Qu.: 89.00 3rd Qu.:172.0 3rd Qu.:31.00 3rd Qu.: 8.000
Max. :1.0000 Max. :1.00000 Max. :8.0000 Max. :108.00 Max. :196.0 Max. :38.00 Max. :120.000

> # Valores NA
> any(is.na(Absenteeism))
[1] FALSE

```

```

> # Convertir los datos a un DataFrame
> Absen<- as.data.frame(scale(Absenteeism))
> # Obtención de medidas estadísticas del DF
> summary(Absen)
Reason.for.absence Month.of.absence Day.of.the.week Seasons Transportation.expense
Min. :-2.2786 Min. :1.84045 Min. :-1.34691 Min. :-1.3892 Min. :-1.54334
1st Qu.: -0.7371 1st Qu.: -0.96742 1st Qu.: -0.64351 1st Qu.: -0.4898 1st Qu.: -0.63224
Median : 0.4487 Median : -0.09438 Median : 0.05988 Median : 0.4096 Median : 0.05482
Mean : 0.0000 Mean : 0.00000 Mean : 0.00000 Mean : 0.0000 Mean : 0.00000
3rd Qu.: 0.8044 3rd Qu.: 0.77865 3rd Qu.: 0.76328 3rd Qu.: 1.3090 3rd Qu.: 0.57758
Max. : 1.0415 Max. : 1.65169 Max. : 1.46668 Max. : 1.3090 Max. : 2.48939

Distance.from.Residence.to.Work Service.time Age Work.load.Average/day Hit.target
Min. :-1.6601 Min. :-2.6350 Min. :-1.45861 Min. :-1.6789 Min. :-3.5953
1st Qu.: -0.9187 1st Qu.: -0.8105 1st Qu.: -0.84121 1st Qu.: -0.6939 1st Qu.: -0.4201
Median : -0.2447 Median : 0.1017 Median : 0.08489 Median : -0.1854 Median : 0.1091
Mean : 0.0000 Mean : 0.0000 Mean : 0.00000 Mean : 0.0000 Mean : 0.0000
3rd Qu.: 1.3729 3rd Qu.: 0.7859 3rd Qu.: 0.54794 3rd Qu.: 0.5819 3rd Qu.: 0.6383
Max. : 1.5077 Max. : 3.7506 Max. : 3.32625 Max. : 2.7496 Max. : 1.4320

Disciplinary.failure Education Son Social.drinker Social.smoker Pet
Min. :-0.2389 Min. :-0.4336 Min. :-0.92756 Min. :-1.1449 Min. :-0.2804 Min. :-0.5659
1st Qu.: -0.2389 1st Qu.: -0.4336 1st Qu.: -0.92756 1st Qu.: -1.1449 1st Qu.: -0.2804 1st Qu.: -0.5659
Median : -0.2389 Median : -0.4336 Median : -0.01722 Median : 0.8723 Median : -0.2804 Median : -0.5659
Mean : 0.0000 Mean : 0.0000 Mean : 0.00000 Mean : 0.0000 Mean : 0.0000 Mean : 0.0000
3rd Qu.: -0.2389 3rd Qu.: -0.4336 3rd Qu.: 0.89312 3rd Qu.: 0.8723 3rd Qu.: -0.2804 3rd Qu.: 0.1927
Max. : 4.1805 Max. : 4.0225 Max. : 2.71380 Max. : 0.8723 Max. : 3.5618 Max. : 5.5028

Weight Height Body.mass.index Absenteeism.time.in.hours
Min. :-1.7880 Min. :-1.51033 Min. :-1.7914 Min. :-0.51941
1st Qu.: -0.7789 1st Qu.: -0.51613 1st Qu.: -0.6247 1st Qu.: -0.36939
Median : 0.3078 Median : -0.35043 Median : -0.3913 Median : -0.29438
Mean : 0.0000 Mean : 0.00000 Mean : 0.0000 Mean : 0.00000
3rd Qu.: 0.7735 3rd Qu.: -0.01903 3rd Qu.: 1.0088 3rd Qu.: 0.08069
Max. : 2.2483 Max. : 3.95777 Max. : 2.6422 Max. : 8.48216

> # Se aplica la medida de distancia para obtener la matriz distancia
> dist_mat <- dist(Absen, method = 'euclidean')

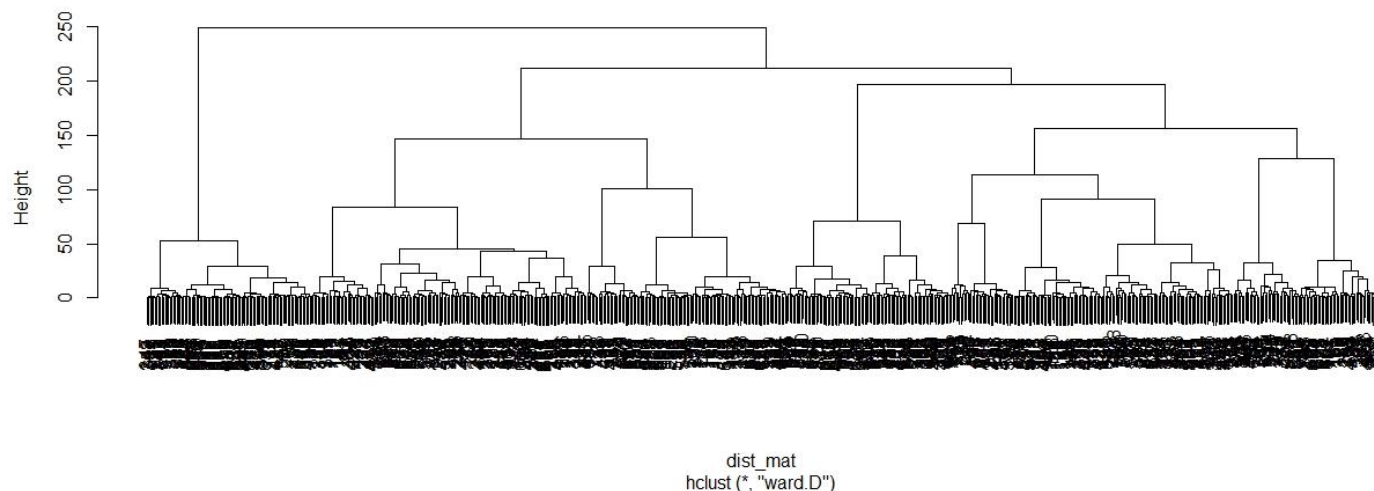
```

```

> # Obtención de los grupos
> grupos <- hclust(dist_mat, method = "ward.D")
> plot(grupos)

```

Cluster Dendrogram

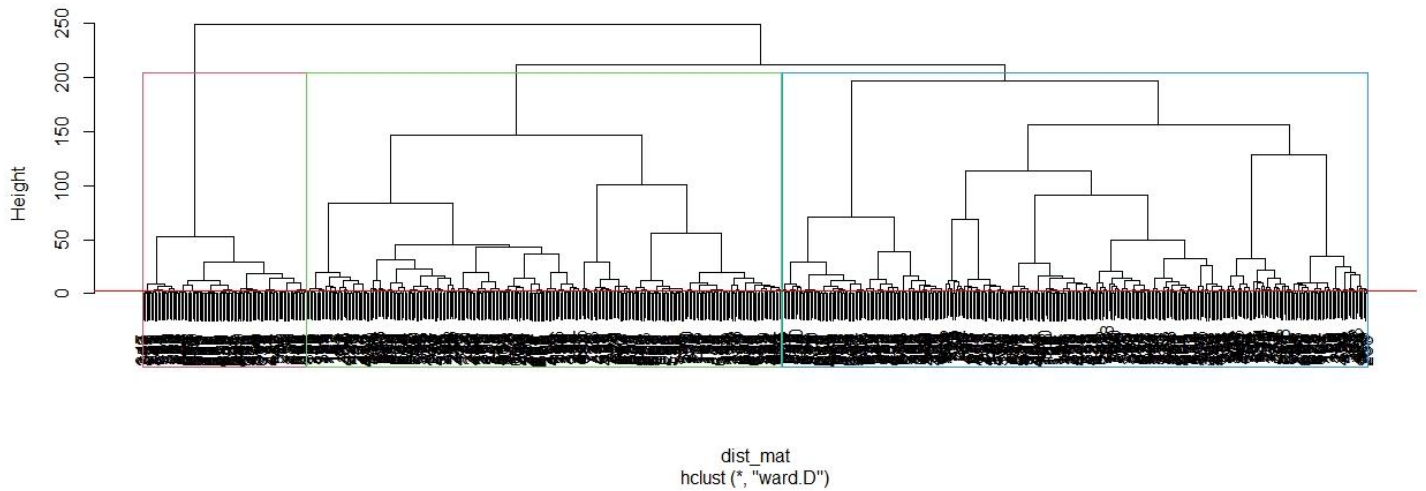


```

> # Trazar la línea de corte y mostrar rectángulo en los grupos
> lineaCorte <- cutree(grupos, k=3)
> plot(grupos)
> rect.hclust(grupos, k = 3, border = 2:6)
> abline(h = 3, col = 'red')

```

Cluster Dendrogram



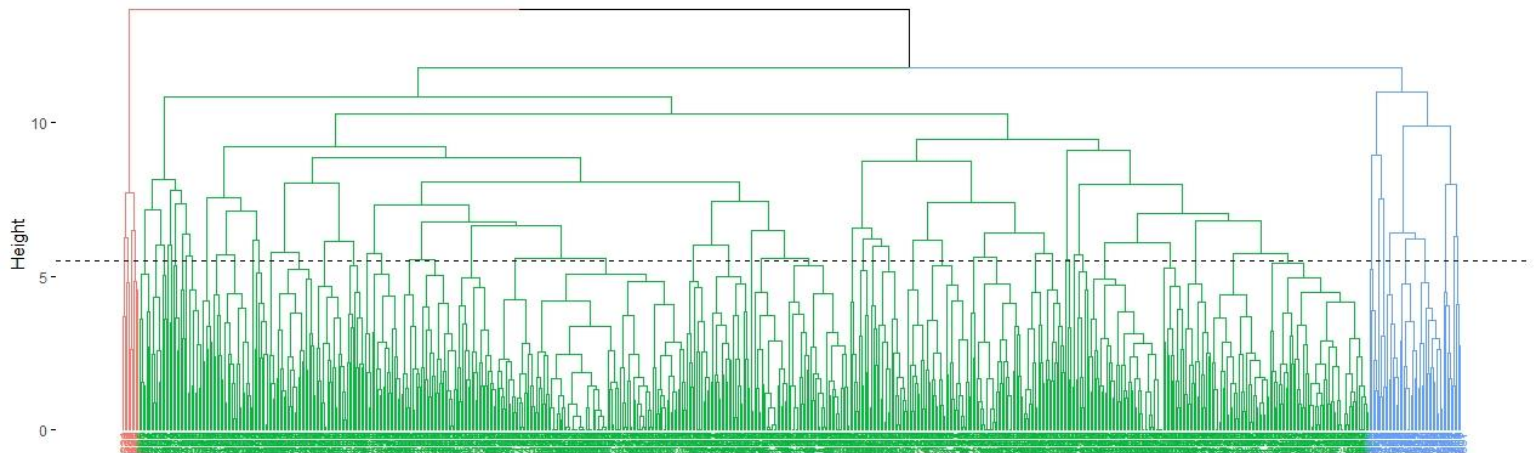
```

> # Extra: aplicamos Hierarchical Clustering
> eucli <- hclust(d = dist(x = Absen, method = "euclidean"), method = "complete")
> fviz_dend(eucli, k = 3, cex = 0.6) + geom_hline(yintercept = 5.5, linetype = "dashed") +
+ labs(title = "Hierarchical clustering", subtitle = "Distancia euclídea, Lincage complete, K=3")

```

### Hierarchical clustering

Distancia euclídea, Lincage complete, K=3





```
> ##### Se aplica algoritmo K-Means #####
> grupoK2 <- kmeans(dist_mat, centers = 2, nstart = 25)
> # Obtención de la estructura de los datos K2
> str(grupoK2)
List of 9
 $ cluster      : Named int [1:740] 1 2 1 2 1 1 2 1 2 1 ...
  .. attr(*, "names")= chr [1:740] "1" "2" "3" "4" ...
 $ centers      : num [1:2, 1:740] 4.68 6.71 7.63 8.09 4.75 ...
  .. attr(*, "dimnames")=List of 2
   .. $ : chr [1:2] "1" "2"
   .. $ : chr [1:740] "1" "2" "3" "4" ...
 $ totss       : num 983248
 $ withinss    : num [1:2] 444989 245162
 $ tot.withinss: num 690151
 $ betweenss   : num 293097
 $ size        : int [1:2] 551 189
 $ iter        : int 1
 $ ifault      : int 0
 - attr(*, "class")= chr "kmeans"
> # Impresión de los grupos
> print(grupoK2)
```

K-means clustering with 2 clusters of sizes 551, 189

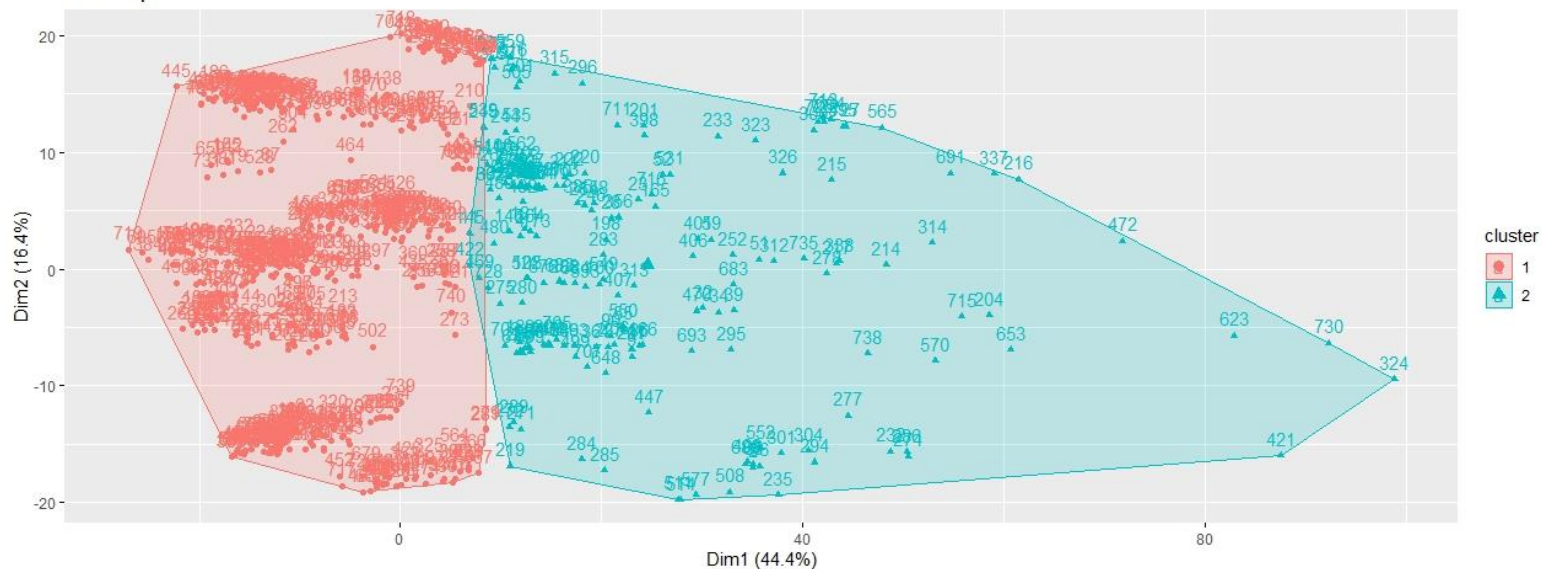
Cluster means:															
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	4.682158	7.629879	4.752851	6.524341	4.667974	4.974704	6.274157	5.628893	7.063976	5.388089	6.148096	6.024581	5.602609	4.938739	4.748076
	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
1	4.945742	4.974704	4.956407	4.878164	5.218329	6.093766	5.376480	7.053346	5.897602	5.074332	6.478972	4.920437	6.935727	4.886424	5.239482
	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45
1	5.232467	7.404675	5.499436	7.495100	5.214136	6.473008	4.921229	5.332608	7.536986	5.848736	5.423938	5.090942	4.899410	5.595981	6.474381
	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60
1	5.092512	4.901107	4.581750	5.700862	5.211401	7.682560	7.176535	4.834173	4.939589	6.993628	7.677956	4.793702	4.854955	7.459612	5.245138
	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75
1	5.067219	5.68053	4.889486	4.921464	7.641026	5.866073	5.054028	4.974010	4.921464	4.761996	5.027117	4.545266	4.845718	4.759119	5.608199
	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90
1	4.633959	4.879493	4.701699	6.32166	5.718040	5.066038	4.913128	5.165309	6.530624	4.698000	7.130984	5.050192	5.591991	4.913128	6.958390
	91	92	93	94	95	96	97	98	99	100	101	102	103	104	105
1	4.906450	6.492149	5.210550	5.848234	5.41057	5.021594	5.311190	4.700488	5.227795	6.780106	5.776001	5.445827	4.978844	5.831643	6.399996
	106	107	108	109	110	111	112	113	114	115	116	117	118	119	120
1	6.282640	5.038293	5.856307	4.977341	6.282640	5.292493	4.944930	5.007006	4.782032	5.082840	4.971566	6.510208	4.821956	5.386909	4.894752
	121	122	123	124	125	126	127	128	129	130	131	132	133	134	135
1	5.036336	5.110805	4.873845	4.967522	4.925973	4.985146	5.141529	5.110805	4.967522	5.863792	4.756459	4.925973	5.386909	4.985146	5.119909
	136	137	138	139	140	141	142	143	144	145	146	147	148	149	150
1	4.833848	4.741508	5.537001	5.31684	5.014529	6.558068	4.785604	5.757420	4.906347	6.169210	6.394456	5.054527	4.839095	5.859900	5.397398
	151	152	153	154	155	156	157	158	159	160	161	162	163	164	165
1	4.855014	5.965791	5.836887	4.791429	4.790481	5.181086	5.686281	6.881707	6.532992	4.645866	5.721297	6.042325	5.420603	6.457096	7.225124
	166	167	168	169	170	171	172	173	174	175	176	177	178	179	180
1	7.155686	5.892299	6.660335	5.196969	6.701693	5.849315	5.111610	6.614463	5.326897	4.902081	5.242288	6.169318	5.098503	5.117889	5.081987
	181	182	183	184	185	186	187	188	189	190	191	192	193	194	195
1	6.551885	5.162498	5.417978	5.163776	4.520384	4.715242	6.501682	4.970108	6.460692	4.984187	5.685619	5.107525	5.642465	4.660357	4.945859
	196	197	198	199	200	201	202	203	204	205	206	207	208	209	210
1	4.729230	5.241663	6.842501	4.683021	6.657018	7.167412	4.847144	5.671208	8.925304	5.314079	6.942594	7.044440	5.452268	5.361893	6.061389
	211	212	213	214	215	216	217	218	219	220	221	222	223	224	225
1	7.041594	5.554537	5.497488	8.439500	8.071086	9.131483	8.120676	5.579919	6.349491	6.731154	5.365217	6.665619	5.568319	5.028719	5.527461
	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
1	5.632170	6.936425	5.521134	6.117013	6.003445	6.081517	8.338372	7.541746	5.646547	7.642082	5.505479	6.486538	6.073726	5.387574	5.049957

	241	242	243	244	245	246	247	248	249	250	251	252	253	254	255
1	5.780267	5.798318	6.566830	6.352933	6.237625	6.849613	6.134068	4.626854	4.833527	5.219425	4.660881	7.531316	4.776297	6.818180	4.758703
	256	257	258	259	260	261	262	263	264	265	266	267	268	269	270
1	5.911833	6.159987	6.138877	5.627539	4.593798	4.922992	5.152160	5.760284	5.661239	5.118609	5.209896	5.579670	5.221165	6.305231	5.550700
	271	272	273	274	275	276	277	278	279	280	281	282	283	284	285
1	6.351227	6.240718	6.080218	8.418354	6.324983	5.857205	8.150119	8.019596	6.154427	6.446161	6.120815	5.985137	6.790330	6.637408	6.786953
	286	287	288	289	290	291	292	293	294	295	296	297	298	299	300
1	8.410891	5.969567	6.694866	6.301154	5.163703	5.731655	5.183581	5.167387	7.959433	7.521298	6.693486	5.612626	5.811680	5.207490	7.922917
	301	302	303	304	305	306	307	308	309	310	311	312	313	314	315
1	7.758854	5.090301	5.973384	7.890987	5.954172	5.954172	5.979645	4.907698	6.175857	5.987765	6.673816	7.765635	7.069873	8.662939	6.540927
	316	317	318	319	320	321	322	323	324	325	326	327	328	329	330
1	6.309540	5.938567	5.293418	5.288650	5.324193	5.790580	4.802341	7.769627	10.96902	5.803040	7.812334	5.363831	5.112102	5.414210	5.043390
	331	332	333	334	335	336	337	338	339	340	341	342	343	344	345
1	5.100308	4.919528	5.096087	5.044811	5.743783	5.795661	8.989779	8.193689	4.994304	5.297940	5.303565	5.307409	5.635887	5.911796	5.900780
	346	347	348	349	350	351	352	353	354	355	356	357	358	359	360
1	5.397678	5.173334	5.320788	6.053205	5.154216	5.532609	5.320788	5.716772	5.435201	5.589130	6.983577	5.152691	4.940234	5.097809	5.767523
	361	362	363	364	365	366	367	368	369	370	371	372	373	374	375
1	6.847896	5.704001	6.738593	5.237169	5.320358	6.021714	6.104081	5.227207	5.422126	4.836980	4.675239	4.942981	4.737062	5.608632	4.868091
	376	377	378	379	380	381	382	383	384	385	386	387	388	389	390
1	4.896805	4.675239	4.932139	4.648180	4.903148	4.772285	4.893795	4.618661	4.408809	6.804380	4.675239	4.903148	5.166814	4.675239	4.903148
	391	392	393	394	395	396	397	398	399	400	401	402	403	404	405
1	4.811520	4.661169	4.852372	5.620095	4.927488	5.326672	6.317565	7.151970	5.005284	6.053245	7.373919	5.202171	5.703317	5.617948	5.601625
	406	407	408	409	410	411	412	413	414	415	416	417	418	419	420
1	7.305506	6.919240	7.591934	4.955536	4.760645	5.597696	4.699241	4.696373	5.593413	5.840707	5.439758	5.086998	5.643691	4.703920	5.661599
	421	422	423	424	425	426	427	428	429	430	431	432	433	434	435
1	10.31650	6.225945	5.591864	4.908873	4.883350	5.712646	4.878302	5.605296	5.655782	4.702700	6.077837	5.592571	6.522795	6.595920	5.837645
	436	437	438	439	440	441	442	443	444	445	446	447	448	449	450
1	4.633635	5.898907	5.856677	4.981683	5.821862	5.655782	4.759621	4.767048	4.675606	4.505562	5.452321	7.025494	5.209830	4.996200	4.509519
	451	452	453	454	455	456	457	458	459	460	461	462	463	464	465
1	4.765261	5.321226	4.933906	4.647310	5.503964	5.097472	5.683841	4.843644	5.829137	5.271893	5.869687	4.876109	5.783393	5.525894	5.977816
	466	467	468	469	470	471	472	473	474	475	476	477	478	479	480
1	5.141287	5.725865	5.000086	6.341202	7.366929	4.833019	9.525587	4.944241	5.826411	6.328836	4.874777	5.727782	4.833019	4.636988	6.359641
	481	482	483	484	485	486	487	488	489	490	491	492	493	494	495
1	4.821984	6.488877	4.743971	4.908180	4.668351	5.526552	4.537143	4.959213	6.373950	5.819663	5.240475	4.752702	4.907838	4.542017	5.681428
	496	497	498	499	500	501	502	503	504	505	506	507	508	509	510
1	5.181780	5.181780	5.338323	6.756784	4.988717	6.298148	5.565689	5.385243	5.163993	6.296882	5.732261	6.462658	7.431820	5.723557	5.030572



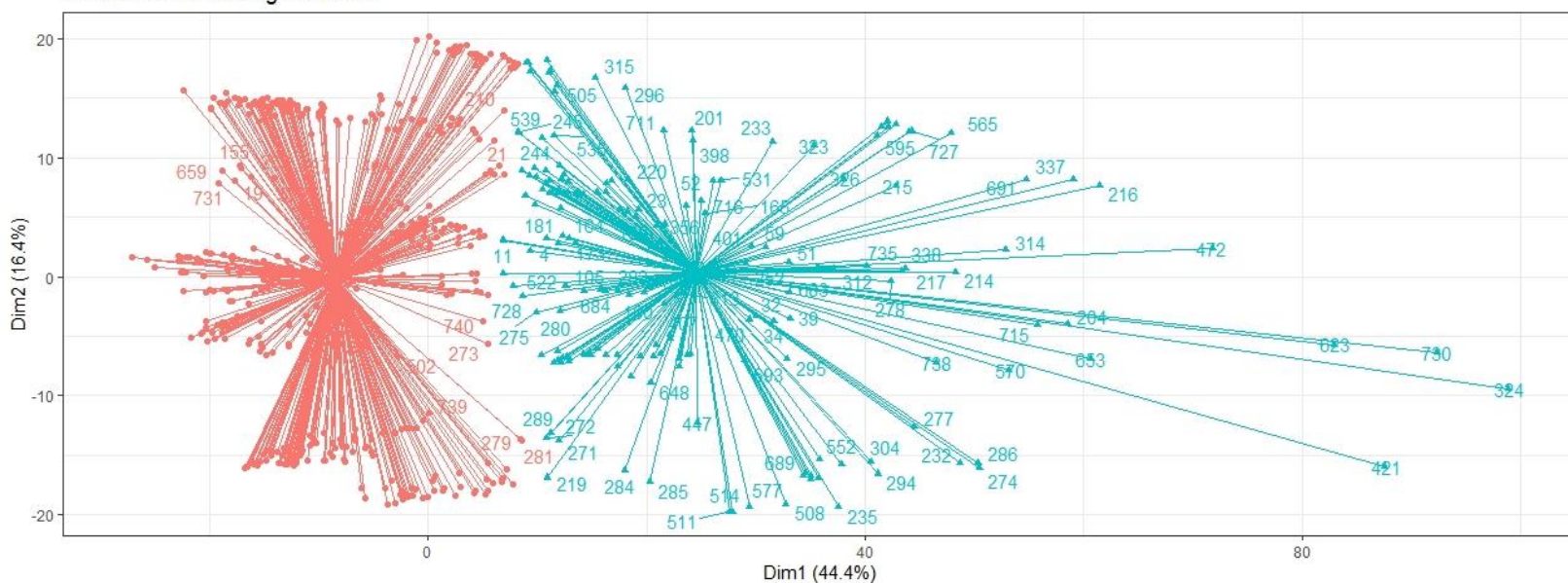


Cluster plot



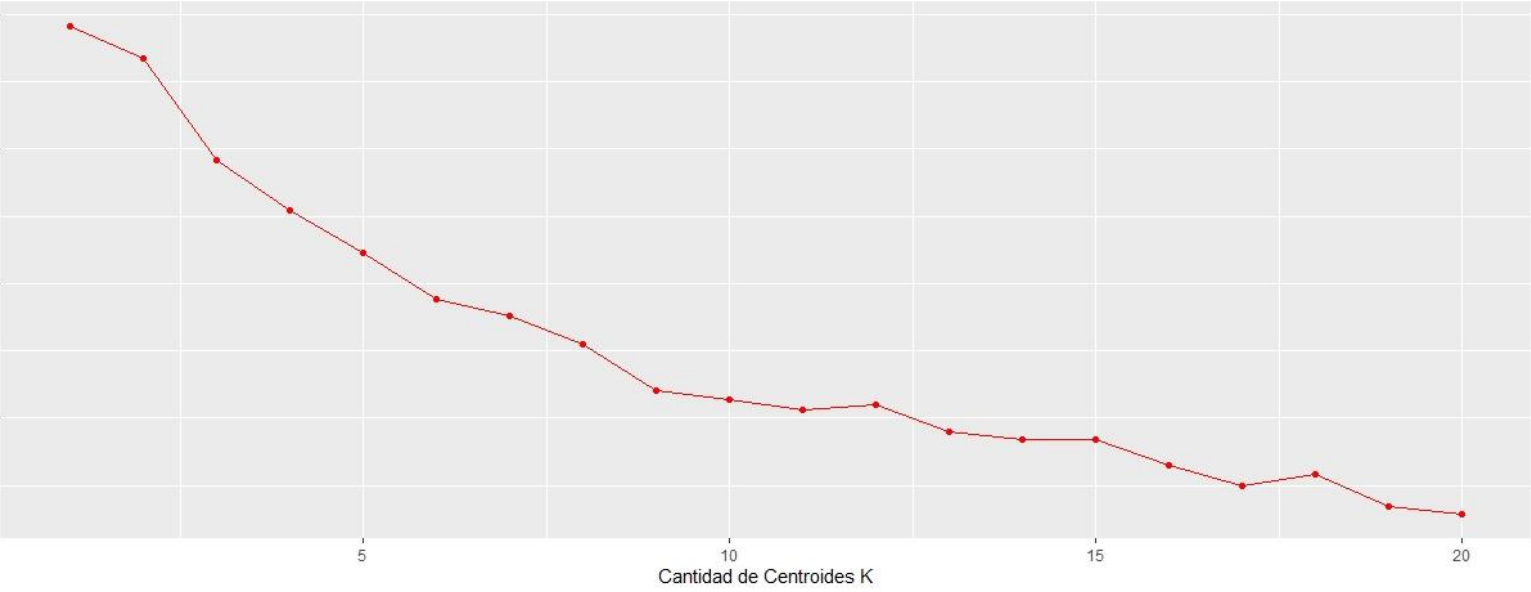
```
> # Resultados de algoritmo K-means
> set.seed(1234)
> fviz_cluster(grupoK2, data = dist_mat, show.clust.cent = TRUE, ellipse.type = "euclid", star.plot = TRUE,
+               repel = TRUE) + labs(title = "Resultados clustering K-means") + theme_bw() + theme(legend.position = "none")
```

Resultados clustering K-means



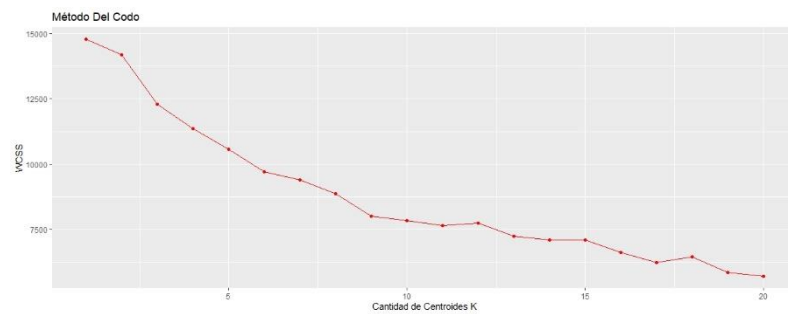
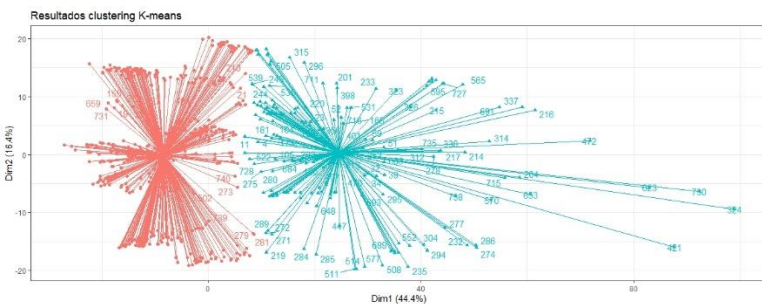
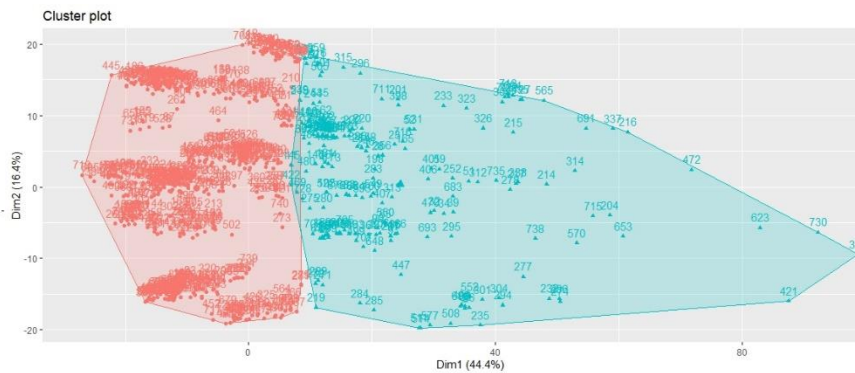
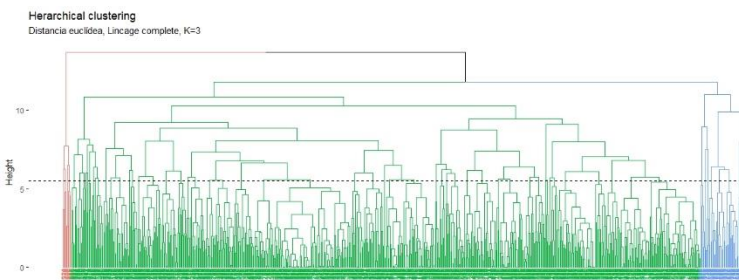
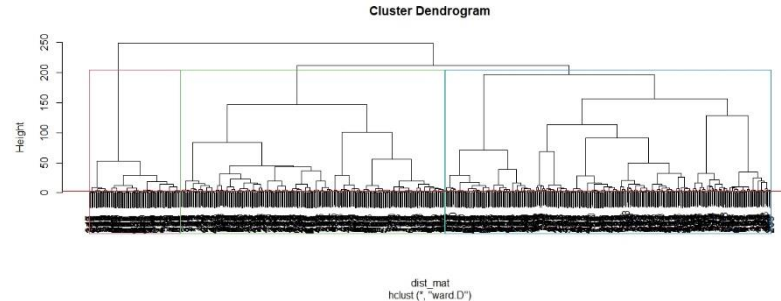
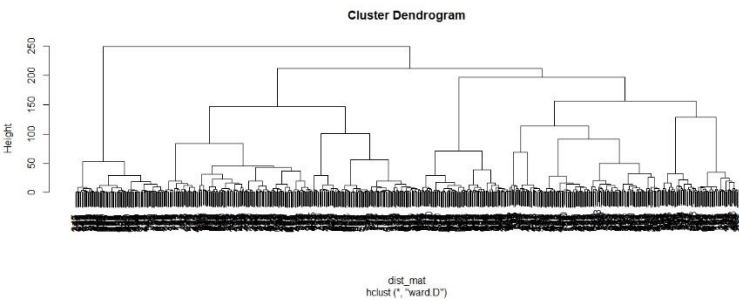
```
> ##### Método del codo #####
There were 16 warnings (use warnings() to see them)
> set.seed(1234)
> wcss <- vector()
> for ( i in 1:20)
+ wcss[i] <- sum(kmeans(Absen,i)$withinss)
> ggplot() + geom_point(aes(x = 1:20, y = wcss), color = 'red') + geom_line(aes(x = 1:20, y = wcss), color = 'red') +
+ ggtitle ("Método Del Codo ") + xlab('Cantidad de Centroides K') + ylab('WCSS')
>
```

Método Del Codo



## Comparativa de los algoritmos

Algoritmo jerárquico	Algoritmo K-Means	Algoritmo del codo
Dentro de esta práctica se utilizó el método aglomerativo y divisivo, en el cual se calculan las distancias entre los grupos existentes y se unen los grupos más similares	Se basa en medidas de distancias entre los grupos en un conjunto de variables cuantitativas, asigna individuos a un número fijo de grupos	Utiliza la distancia media de las observaciones a su centroide. Es decir, se fija en las distancias intra-cluster.



Los 3 Algoritmos como vemos son similares, ya que al codificarlo podemos obtener de manera eficiente, son capaces de encontrar las semejanzas de los datos y producir de esa manera grupos que se convierten en gráficas como lo vimos anteriormente.

**Conclusión:** Al termino de esta práctica se pudo aprender y comprender como implementar el Análisis de Clúster, con algoritmos que se pueden aplicar a conjuntos de datos enormes para entonces poder reducirlos, aunque bien, la visualización dentro de los gráficos resultaría muy difícil, por lo tanto, estos sirven como un método sencillo de clasificación cuando no se tiene información previa o títulos de estos mismos.

