



INSTITUTO POLITECNICO NACIONAL ESCUELA SUPERIOR DE CÓMPUTO

“Practica Clustering”

Integrantes:

- Tinoco Videgaray Sergio Ernesto
- Castro Mendieta Fernando
- Porras Zúñiga Braulio Gael

Grupo: 5BV1

Materia: Aprendizaje De Máquina



I.	Introducción:	3
	K-Means	3
	Jerárquico	4
	Método del Codo	5
	Dendrogramas	5
II.	Desarrollo:	6
	• En Python	6
	• En R	10
III.	Conclusiones:	14
IV.	Referencias:	14

I. Introducción:

En el área del aprendizaje de maquina existen múltiples herramientas que nos permiten procesar los datos recopilados de tal forma que nos permitan obtener información que a simple vista no nos es muy notoria.

Algunas de estas herramientas nos permiten utilizar modelos matemáticos y estadísticos con el fin de cuantificar los datos y visualizarlos de manera grafica.

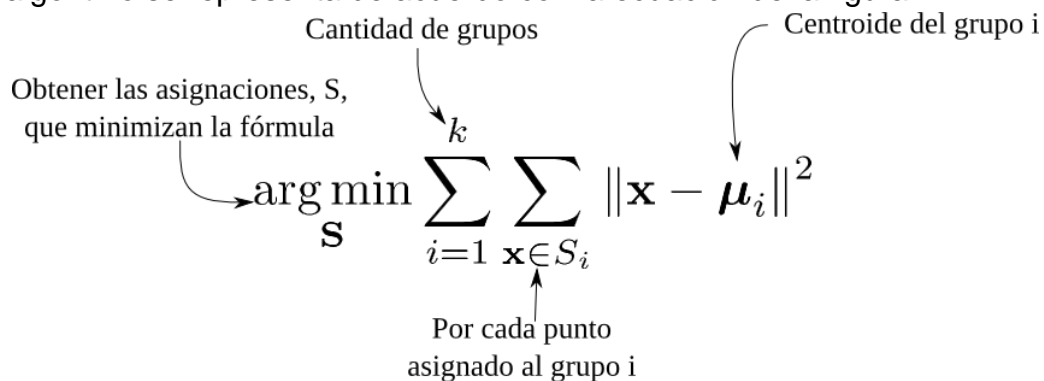
Un ejemplo de estas herramientas son los algoritmos de agrupamiento, también conocidos como “algoritmos de clusterización”.

Como ejemplo de estos algoritmos de agrupación tenemos el algoritmo de K-medias o también conocido como K-means, propuesto por MacQueen en el año 1967.

K-Means

K- means es un algoritmo iterativo que tiene como objetivo generar una partición de un conjunto de n observaciones en k grupos. En el cual, cada grupo está compuesto por la media (de ahí el nombre) de los puntos que lo componen. El representante de cada grupo se le denomina centroide, el cual se localiza en el centro geométrico de cada grupo o “cluster”. K, es un parámetro que se debe definir arbitrariamente. Después de asignar los centroides, estos se reasignan a la ubicación promedio de todos los datos asignados a él, esto de forma iterativa de acuerdo con las nuevas posiciones de los centroides.

Este algoritmo se representa de acuerdo con la ecuación de la figura 1.1



The diagram shows the mathematical formula for the K-Means objective function, $\arg \min_S \sum_{i=1}^k \sum_{\mathbf{x} \in S_i} \|\mathbf{x} - \mu_i\|^2$, with several annotations: an arrow from 'Cantidad de grupos' points to the variable k ; an arrow from 'Obtener las asignaciones, S, que minimizan la fórmula' points to the $\arg \min_S$ term; an arrow from 'Centroide del grupo i' points to the μ_i term; and an arrow from 'Por cada punto asignado al grupo i' points to the inner summation $\sum_{\mathbf{x} \in S_i}$.

Figura 1.1
Ecuación para
generar los k-grupos



Como se pudo observar en la figura 1, el algoritmo de clusterización K-means busca encontrar un valor mínimo, el cual corresponde a la distancia del centroide con los puntos asignados a su clúster, de tal forma que los puntos de un clúster dado siempre se encuentren a la menor distancia posible de su centroide.

Otro algoritmo que nos permite realizar el agrupamiento de datos por medio de Clusters son los conocidos algoritmos jerárquicos, los cual tienen por objetivo agrupar Clusters para formar uno nuevo o bien, separar algún cluster ya existente para dar origen a otra cantidad de Clusters, de tal forma que, si sucesivamente se va efectuando este proceso de aglomeración o división, se minimice alguna distancia o bien se maximice alguna medida de similitud.

Jerárquico

Los métodos jerárquicos se subdividen en aglomeratelo y divisivos.

- Los métodos aglomerativos, también conocidos como ascendentes, comienzan el análisis con tantos grupos como individuos haya, y a partir de estas unidades iniciales se van formando los grupos correspondientes, esto de forma ascendente, hasta que al final del proceso todos los casos tratados están englobados en un mismo clúster.
- Los métodos divisivos, también llamados descendentes, constituyen el proceso inverso al anterior. Es decir, comienzan con un único clúster que engloba a todos los casos tratados y, a partir de este cluster inicial, por medio de sucesivas divisiones, se van formando Clusters cada vez más pequeños, de tal forma que al final del proceso se tienen tantos Clusters como individuos.

Estos métodos de agrupación no son eficientes para todos los casos, ya que algunas veces se procesan datos de diferentes categorías como textos, imágenes, audios, etc. Los cuales podrían impactar en el desempeño del algoritmo de clusterización de tal forma que los resultados obtenidos no sean los más eficientes en la práctica. Debido a ello, se han desarrollado distintos modelos que nos permiten realizar una preevaluación de nuestro algoritmo por medio de métodos estadísticos que permutan los datos de entrada como lo es el número de Clusters o las distancias entre los grupos con el fin de encontrar una posible solución “óptima” para estos algoritmos.

Algunos de estos modelos se les conoce como métodos de Análisis de Cluster. Dichos modelos nos permiten elegir un número adecuado de Clusters para agrupar los datos de manera eficiente. A continuación, se muestran algunos ejemplos.

Método del Codo

Este método utiliza los valores de la inercia obtenidos tras aplicar el K-means a diferente número de Clusters (desde 1 a N Clusters), siendo la inercia la suma de las distancias al cuadrado de cada objeto del Cluster a su centroide como se muestra en la ecuación de la figura 1.2.

$$Inercia = \sum_{i=0}^N \|x_i - \mu\|^2$$

Figura 1.2
Ecuación para generar las distancias
dentro de los clusters

Una vez obtenidos los valores de la inercia tras aplicar el K-means de 1 a N Clusters, representamos en una gráfica lineal la inercia respecto del número de Clusters. En esta gráfica se debería de apreciar un cambio brusco en la evolución de la inercia, teniendo la línea representada una forma similar a la de un brazo y su codo. El punto en el que se observa ese cambio brusco (el punto crítico) en la inercia nos dirá el número óptimo de Clusters a seleccionar para ese data set. Dicho de otra manera: el punto que representaría al codo del brazo será el número óptimo de Clusters para el data set.

Dendrogramas

Un dendrograma es un gráfico en forma de árbol que organiza y agrupa los datos en subcategorías de acuerdo con su similitud *figura 1.3*; dada por alguna medida de distancia como la Euclideana o la distancia Coseno. Los objetos similares se representan en el dendrograma por medio de un enlace cuya posición está dada por el nivel de similitud entre los objetos o grupos de objetos. Por lo que esta herramienta grafica resulta muy útil para estudiar las agrupaciones de objetos; o bien, para estudiar los Clusters que pueden darse en un data set.

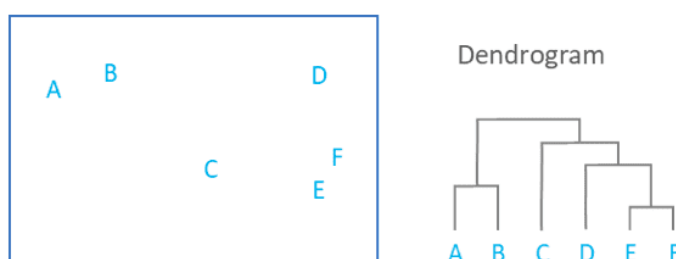


Figura 1.3
Representación de un dendrograma.



En esta práctica se van a implementar los algoritmos de análisis de Clusters para un dataset dado y de igual forma se van a implementar los algoritmos de clusterización Jerárquico y K-means en los lenguajes de programación R y Python. Para efectos de esta practica se va a trabajar con un dataset que nos proporciona la composición química de algunas bebidas alcohólicas, así como sus características físicas.

Alcohol	Malic_Acid	Ash	Ash_Alcanity	Magnesium	Total_Phenols	Flavanoids	Nonflavanoid_Phenols	Proanthocyanins	Color_Intensity
14.23	1.71	2.43	15.6	127	2.80	3.06	0.28	2.29	5.64000
13.20	1.78	2.14	11.2	100	2.65	2.76	0.26	1.28	4.38000
13.16	2.36	2.67	18.6	101	2.80	3.24	0.30	2.81	5.68000
14.37	1.95	2.50	16.8	113	3.85	3.49	0.24	2.18	7.80000
13.24	2.59	2.87	21.0	118	2.80	2.69	0.39	1.82	4.32000
14.20	1.76	2.45	15.2	112	3.27	3.39	0.34	1.97	6.75000
14.39	1.87	2.45	14.6	96	2.50	2.52	0.30	1.98	5.25000
14.06	2.15	2.61	17.6	121	2.60	2.51	0.31	1.25	5.05000
14.82	1.64	2.17	14.0	97	2.80	2.98	0.30	1.08	5.30000

Figura 1.4
Dataset de la composición química de
bebidas alcohólicas.

II. Desarrollo:

- Implementación en Python

Para la primera parte del desarrollo se van a importar las bibliotecas necesarias para la implementación de los algoritmos.

```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.cluster import KMeans
import scipy.cluster.hierarchy as sch
```

Primeramente, se importa el archivo del dataset correspondiente.

```
dataset=pd.read_csv("dataset.csv")
x=dataset.iloc[:,[0,9]].values
```

Algoritmo Jerárquico.

Para implementar el algoritmo jerárquico se implemento el algoritmo para generar el dendograma y así visualizar mejor la cantidad de Clusters de manera óptima.

```
import scipy.cluster.hierarchy as sch
dendogram=sch.dendrogram(sch.linkage(x,method="ward"))
#Tipo de enlace
plt.title("Dendograma")
plt.xlabel("Precio")
plt.ylabel("Compras")
plt.show()
```

Para el algoritmo jerárquico se consideraron utilizar 3 clusters haciendo una evaluación con el dendograma *figura 2.1*.

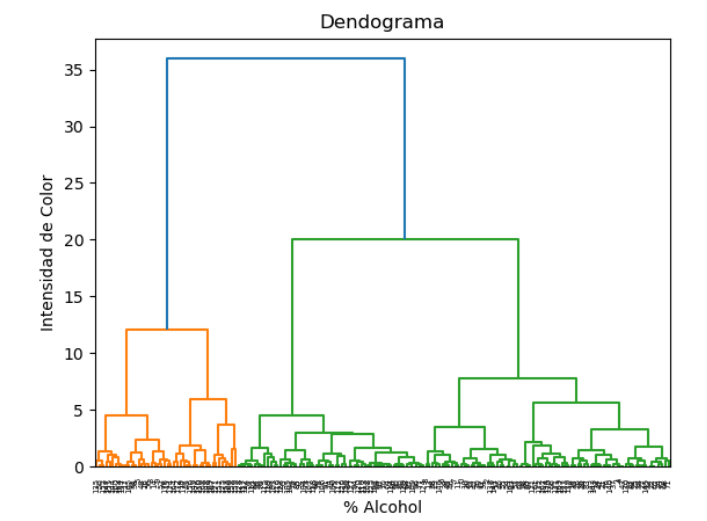


Figura 2.1
Dendograma generado en Python.

Posteriormente se entrena al modelo jerárquico *figura 2.2*.

```
#Ajustar el grupo jerarquico al conjunto de datos
from sklearn.cluster import AgglomerativeClustering
hc=AgglomerativeClustering(n_clusters=3,affinity="euclidean",linkage
="ward")
y_hc=hc.fit_predict(x)
```

Resultados.

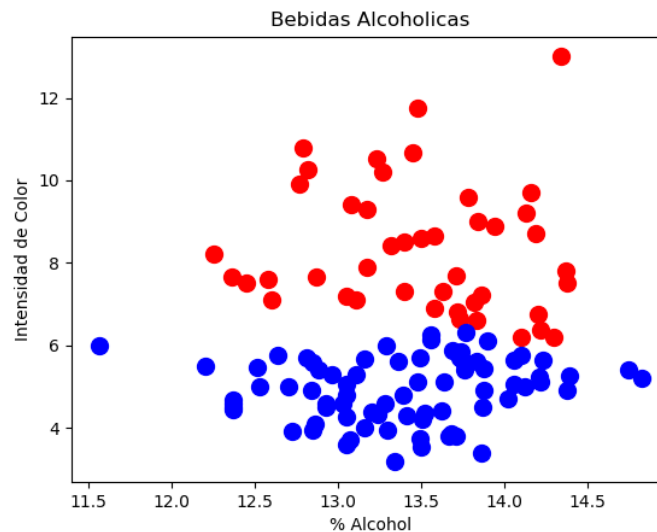


Figura 2.2
Clustering jerárquico
generado en Python.

Algoritmo K-Means

Para el algoritmo de K-means se va a implementar el método del codo de Jambú.

Para ello se crea una lista con el fin de almacenar la suma del cuadrado de las distancias dentro de Clusters.

```
wcss=[]          #Lista para distancias

for i in range(1,11):
    kmeans=KMeans(n_clusters=i,init="k-
means++",max_iter=300,n_init=10,random_state=0)
    kmeans.fit(x)    #Ajustar la matriz de características
    wcss.append(kmeans.inertia_)    #Agrega el atributo en la lista wcss
```

Posteriormente se plotean las distancias en un intervalo de 1 a 10 *figura 2.3*.

```
#Grafica de codo
plt.plot(range(1,11),wcss)
plt.title("Metodo del codo")
plt.xlabel("Numero de clusters")
plt.ylabel("WCSS(k)")
plt.show()
```


Resultado.

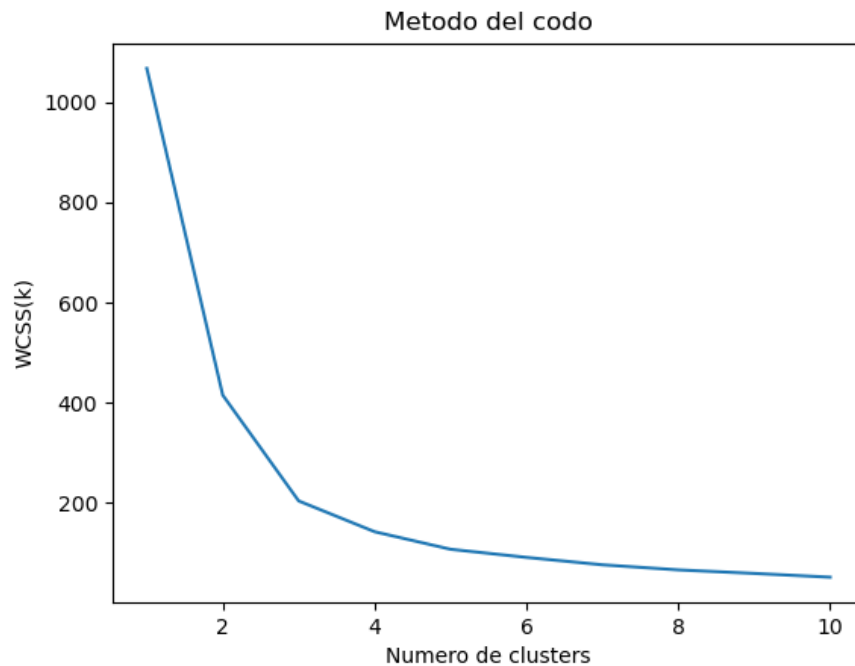


Figura 2.3
Codo de Jambú generado en python

Como se puede apreciar en la gráfica de codo, el punto crítico más bajo se encuentra en la posición 3, por lo que el número de Clusters óptimo para el algoritmo son 3 Clusters.

Se entrena el modelo para 3 clusters.

```
kmeans=KMeans(n_clusters=3,init="k-  
means++",max_iter=300,n_init=10,random_state=0)  
y_kmeans=kmeans.fit_predict(x)
```

Finalmente se grafican los Clusters figura 2.4.

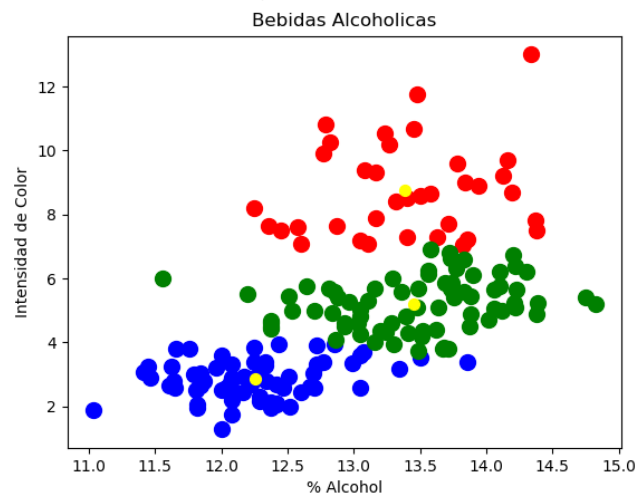


Figura 2.4
Clustering con 3 Clusters generado en Python.

- Implementación en R.

Primeramente, se importa el archivo del dataset correspondiente.

```
#Importar datos
dataset <- read.csv("C:/Users/Sergio/Desktop/Practica2/dataset.csv")
View(dataset)
```

Posteriormente se carga el dataframe a utilizar en una variable x.

```
x=dataset[,c(1,10)]
```

Jerárquico Aglomerativo.

Se crea una instancia de hclust para generar el dendograma por medio de la distancia Euclideana

```
#Utilizar el dendograma para encontrar el numero optimo de grupos
dendogram=hclust(dist(x,method="euclidean"),
                  method = "ward.D")
```

Se grafica el dendograma generado.

```
plot(dendogram,
     main="Dendograma",
     xlab="% Alcohol",
     ylab="Intensidad de Color")
```

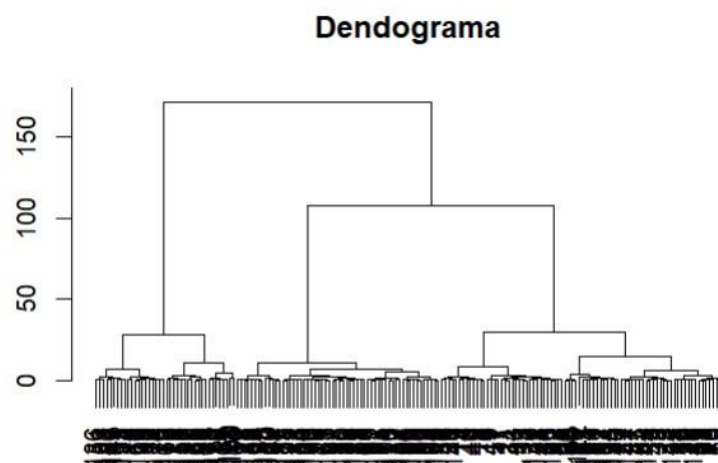


Figura 2.5
Dendograma generado en R.

Se entrena el modelo para el algoritmo jerárquico.

```
#Ajustar el agrupamiento jerarquico al dataset
hc=hclust(dist(x,method="euclidean"),
          method = "ward.D")
y_hc=cutree(hc,k=3)
```

Finalmente se grafican los Clusters.

```
#Visualizar los clusters
#install.packages("cluster")
library(cluster)
clusplot(x, #Variable que contiene los datos
         y_hc, #variable que contiene a que grupo pertenece cada
         lcohólicas
         lines=0,
         shade=TRUE,
         color=TRUE,
         labels=5,
         plotchar=FALSE,
         span=TRUE,
         main="Bebidas lcohólicas",
         xlabel="% Alcohol",
         ylabel="Intensidad de Color")
```

Resultados.

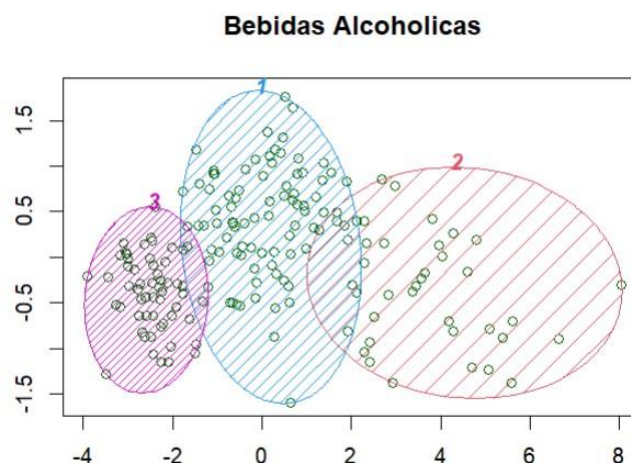


Figura 2.6
Clustering con 3 clusters en R.

Algoritmo K-Means

Para K-means se aplicará el método del codo de Jambú para obtener el mejor número de Clusters

```
#Aplicar el metodo del codo para poder obtener el numero de clusters
```

```
(k)
set.seed(5)
wcss=vector() #Lista de distancias
for (i in 1:10)
{
  wcss[i]<-sum(kmeans(x,i)$withinss)
}
```

Posteriormente se grafica el codo de Jambú generado.

```
#Graficar el codo
plot(1:10,wcss,type="b",
     main="Metodo del codo", xlab = "% Alcohol", ylab="Intensidad de Color")
```

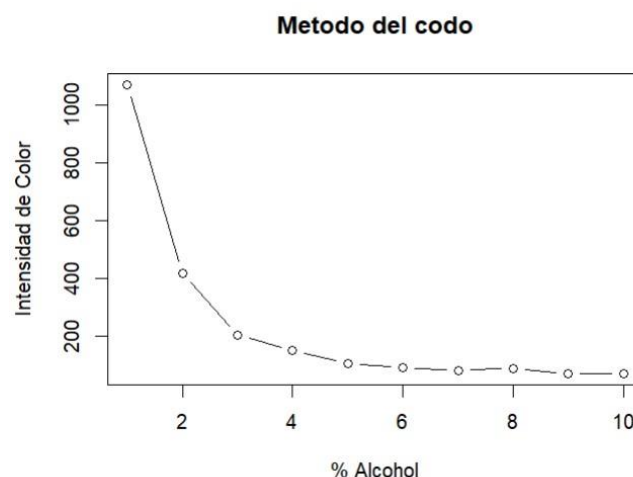


Figura 2.7
Codo de Jambú generado en R.

Como se puede observar la gráfica muestra un punto de inflexión en 3 lo que implica un mejor Clustering de 3 grupos.

Se entrena el algoritmo con 3 clusters.

```
#Aplicar k-means con k optimo
set.seed(5)
kmeans=kmeans(x,3,iter.max = 300,nstart = 10)
```

```
#Visualizacion de los clusters
#install.packages("cluster")
library(cluster)
clusplot(x, #variable que contiene los datos
         kmeans$cluster, # A que grupo pertenece)
         lines=0, #Para puntos
         shade=TRUE, #Sombrea los puntos
         color=TRUE, #Aplica color
         labels=10, #Etiqueta los puntos
         plotchar=FALSE, #Simbolos iguales para todos los grupos
         span=TRUE, #Aparece una elipse que delimita el grupo
         main="Bebidas Alcoholicas", #Titulo
         xlab="% Alcohol", #Titulo del eje x
         ylab="Intensidad de Color") #Titulo del eje y
```

Resultado.

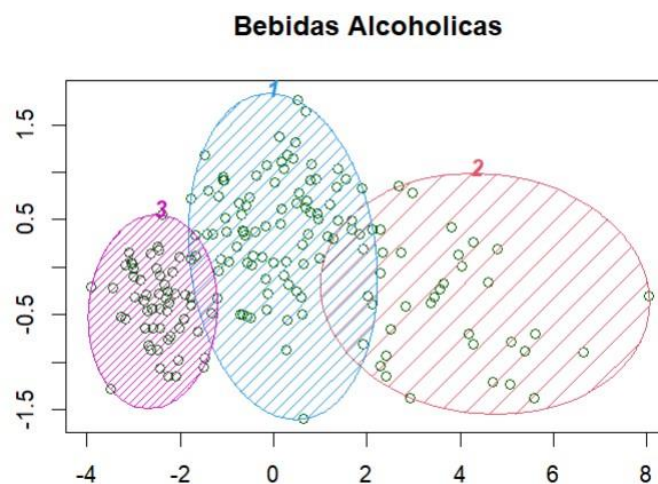


Figura 2.8
Clustering con 3 clusters en R.



III. Conclusiones:

Aunque sería ideal que las máquinas pudieran interpretar las gráficas de análisis de Clustering con el fin de determinar el número óptimo de grupos para Clusterizar, la realidad es que muchos de estos algoritmos requieren la inspección y el análisis de una persona capacitada para interpretar las graficas y determinar de forma “intuitiva” estos parámetros con base en los valores obtenidos.

De igual forma se observo un mejor desempeño con el algoritmo de K-Means para esta practica en cuanto a tiempo de ejecución lo que bien no necesariamente implica que este algoritmo resulte mejor para cualquier tipo de casos, pero si para un caso en el que se cuentan con muchos datos.

IV. Referencias:

- [1] *Definición / K-medias.* (2016). Recuperado 28 de octubre de 2022, de http://163.10.22.82/OAS/Agrupamiento_Kmedias/definicin.html

- [2] Richard. (2017, 2 junio). *Selección del número óptimo de Clusters.* Jarroba. <https://jarroba.com/seleccion-del-numero-optimo-clusters/>

- [3] *Wine Dataset for Clustering.* (2020, 29 abril). Kaggle. <https://www.kaggle.com/datasets/harrywang/wine-dataset-for-clustering>