**1)Aprendizaje supervisado: función que establece una correspondencia entre entradas y salidas deseadas del sistema.**

Ejemplo: problema de clasificación donde el sistema trata de etiquetar (clasificar) una serie de vectores entre varias categorías(clases)

Veremos las técnicas de PCA (componentes principales) que se usan para reducir el número de variables mediante combinaciones. Suelen utilizarse en ese ámbito para aumentar el rendimiento al reducirse el número de variables con las que tiene que tratar el algoritmo que corresponda.

**2)Aprendizaje no supervisado. el proceso de modelado se lleva a cabo sobre un conjunto de ejemplos formado tan sólo por entradas al sistema. No se tiene información sobre las categorías de esos ejemplos.**

En este caso, el sistema tiene que ser capaz de reconocer patrones para poder etiquetar las nuevas entradas.

**De los no supervisados veremos Reglas de Asociación, Patrones Secuenciales y Agrupamiento(Clustering).**

**A)Reglas de Asociación: es un método que permite descubrir relaciones interesantes entre las distintas variables de un gran conjunto de datos. Análisis de cesta de la compra (market basket analysis13**

Las Reglas de Asociación se aplican básicamente sobre conjuntos de datos que se denominan como transaccionales

Ítems, transacciones , id transacciones, conjunto de ítems

**Teoría de las Reglas de Asociación**

**X => Y x,y son subconjuntos de elementos de I (itemsets) y no hay elementos comunes entre ellos**

𝑋,𝑌 ⊆𝐼 y 𝑋 ∩𝑌=0

**Todas las Reglas de Asociación están compuestas por dos itemsets (en este caso X,Y) que tienen al menos un elemento (ítem)**

Al itemset que está en la izquierda de la Regla (X) se le denomina **antecedente o LHS (del inglés Left-Hand-Size)** mientras que el que está en la derecha se le denomina **consecuente o RHS (Right-Hand-Size).**

**Normalmente se suele fijar un umbral mínimo para esas medidas que deberán de cumplir las Reglas que finalmente serán consideradas**

Las medidas de interés más utilizadas son **soporte (support)** y **confianza (confidence),** pero no son las únicas como veremos a continuación

**supp(X) = nX / n**

**Soporte**: Es la proporción de transacciones en la base completa (D) que contienen el itemset X. nX es el número de transacciones en los que aparece el itemset X y n el número total de transacciones.

**𝑐𝑜𝑛𝑓(𝑋=>𝑌) = 𝑠𝑢𝑝𝑝(𝑋∪𝑌)/ 𝑠𝑢𝑝𝑝(𝑋) siendo 𝑠𝑢𝑝𝑝(𝑋 ∪𝑌) el soporte de la unión de los elementos de X y de Y.**

**Confianza**: es la proporción de transacciones que conteniendo el itemset X también contienen el itemset Y.

**𝑙𝑖𝑓𝑡(𝑋=>𝑌)=𝑠𝑢𝑝𝑝(𝑋=>𝑌)𝑠𝑢𝑝𝑝(𝑋) ×𝑠𝑢𝑝𝑝(𝑌)=𝑐𝑜𝑛𝑓(𝑋=>𝑌)𝑠𝑢𝑝𝑝(𝑌)=𝑃(𝑋 ∪𝑌)𝑃(𝑋)𝑃(𝑌)**

**Lift**: es el ratio del soporte real observado frente al que se daría si X e Y fueran probabilísticamente independientes**.**

**Lift = 1 implicaría que la probabilidad de ocurrencia del antecedente y el consecuente en dicha Regla es independiente la una de la otra. Cuando dos eventos son independientes no hay ninguna Regla que relacione ambos eventos.**

**Lift > 1 indica que hay grado de dependencia entre antecedente y consecuente. Cuanto mayor sea el lift de una Regla podríamos decir que más relacionados están el antecedente y el consecuente relacionando de alguna forma el soporte del antecedente y el consecuente apareciendo juntos y la confianza de dicha Regla.**

**Implementación de apriori en R. Se reduce a los siguientes pasos principales:**

• Fijar un soporte y confianza mínima

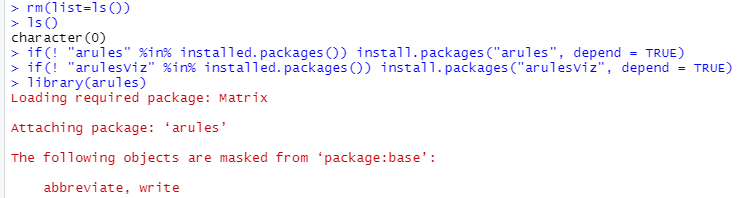
• Quedarse con los itemsets que cumplan la condición de soporte mínimo utilizando la norma de Agrawal and Srikant

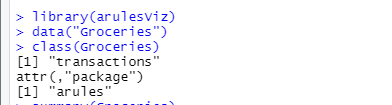
• Se generan todas las Reglas de Asociación posibles con los itemsets resultado del punto anterior y se calcula su confianza

• Se retiran del resultado final las Reglas que no cumplen con el criterio de confianza mínima

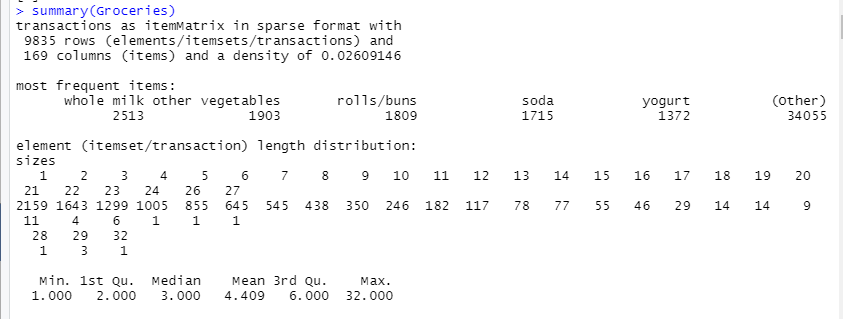
El paquete de referencia en R es Arules

Cargamos librerías y el dataset “Groceries” con las transacciones que representan un grupo de compras y que se encuentra en el paquete arules

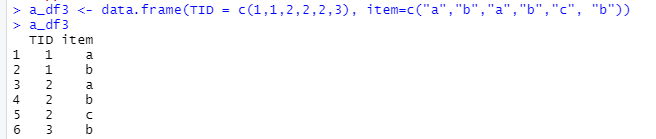


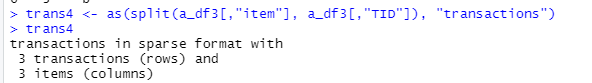


Con summary() vemos un resumen de lo que contiene.

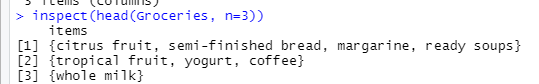


Para tener los datos en formato “transactions”

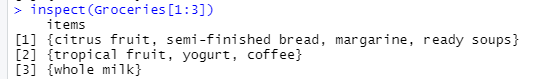




Utilizamos para ello la función inspect() del paquete arules junto con la función head() para limitar el número de registros a mostrar o la notación de vectores típica de R con

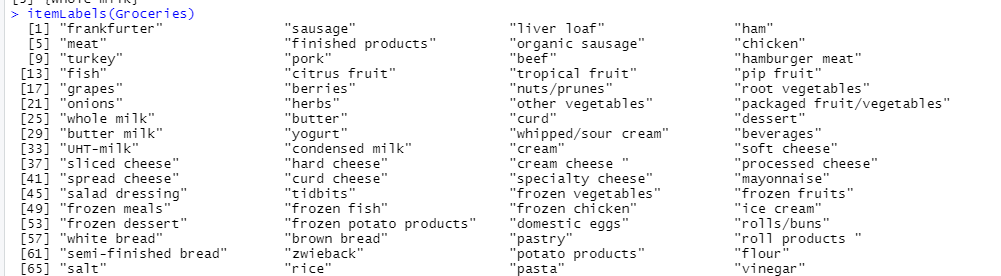


o



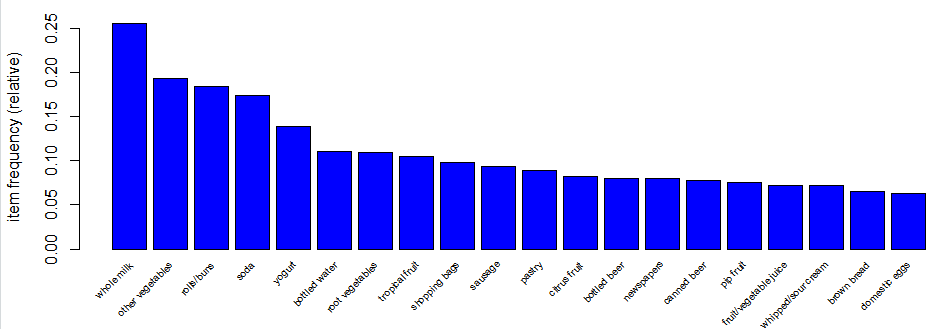
Hacemos breve exploratorio para ver las principales características de nuestro dataset.

. Conjunto de elementos con **itemLabels(Groceries)**



Vemos la gráfica de la distribución de los elementos en el dataset



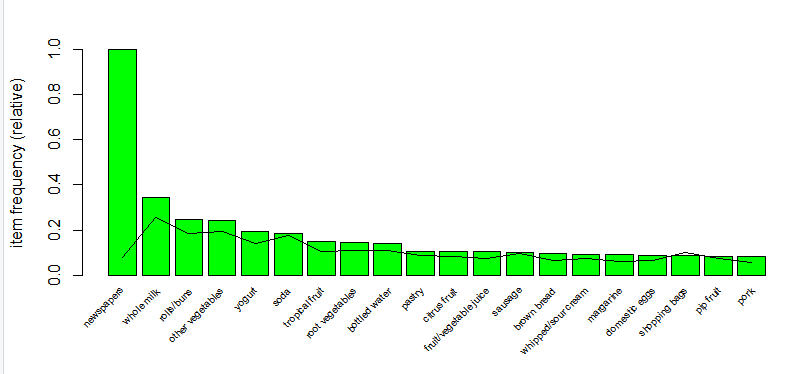


Si queremos filtrar las compras en las que se adquiere un producto en particular podemos utilizar la función subset() con alguna condición

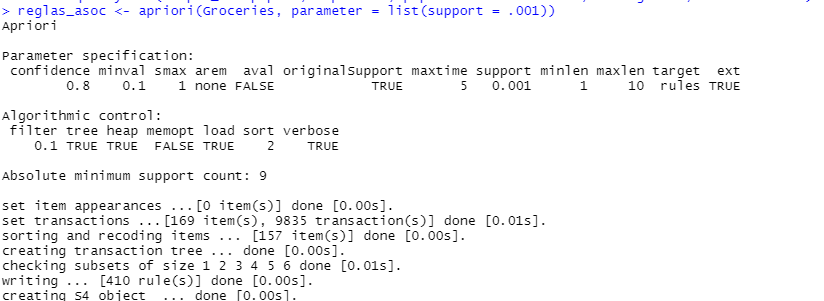


Ver una gráfica de las distribuciones de productos en el subconjunto de compras en las que se adquiere periódicos frente a lo que ocurre en el total de compras

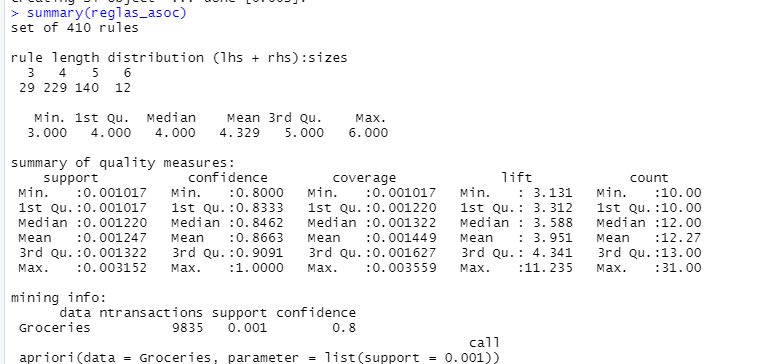


****

Hasta ahora sólo hemos explorado el dataset, ahora vamos a aplicar la **función apriori()** **para obtener las Reglas de Asociación**. Recordemos que tenemos que fijar al menos un soporte mínimo. En este caso vamos a quedarnos con las asociaciones que ocurran al menos en 10 compras del dataset completo que incluye unas 10000. Por tanto **minsup=0.001**

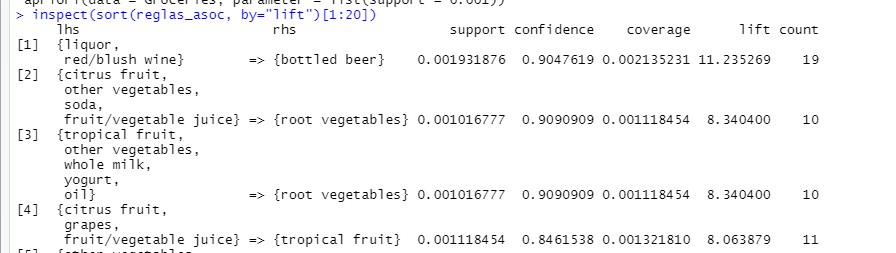
****

A través de la función summary() nos da información muy interesante del conjunto de reglas

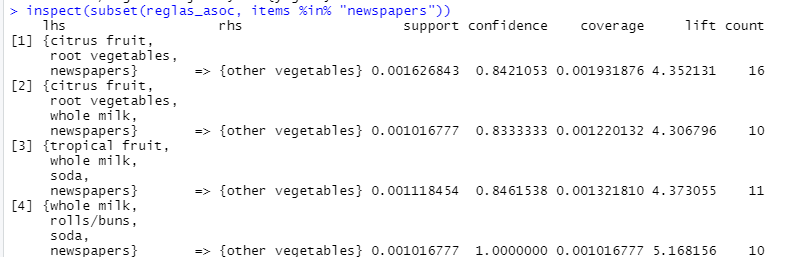


Hay 410 reglas que cumplen con la condición del soporte mínimo, la distribución del tamaño de las reglas (número de productos) y que por defecto aplica la condición de confianza mínima utilizando un valor de 0.8.

Podemos inspeccionar las **reglas**, en este caso las vamos a **ordenar (sort()) por lift** para que salgan las más relevantes arriba y nos quedamos con las 20 primeras



**Si queremos quedarnos con las reglas que cumplen los criterios mínimos de soporte y confianza y además incluyan algún producto en particular (newspapers) la función subset() también funciona con Reglas**

****

**En este caso sólo cuatro reglas que asocian newspapers con otros productos cumplen con los criterios de soporte y confianza mínima**

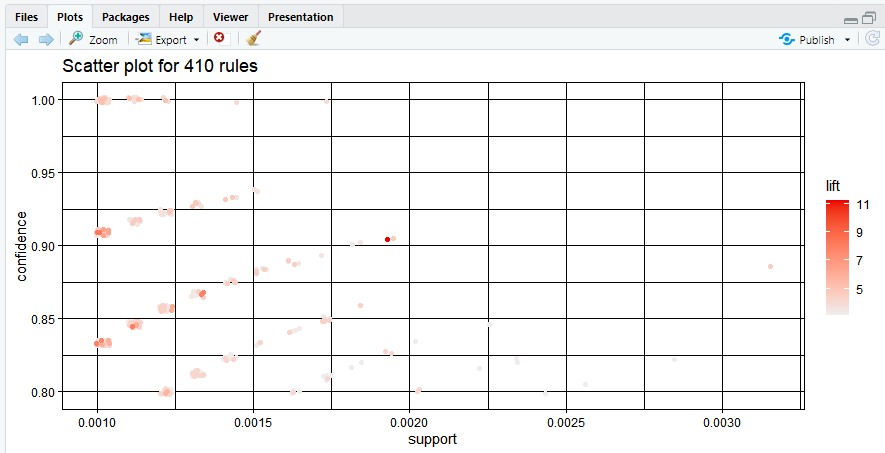
**También se podrían seleccionar las Reglas que cumplen con los criterios de soporte y confianza, pero que en este caso no contengan un determinado artículo. Simplemente añadiendo a la condición de la función subset() el símbolo de negación en R, la exclamación (!)**

****

**Si teníamos 410 reglas en total y había 4 conteniendo newspapers, ahora tenemos 406 que no contienen dicho producto**

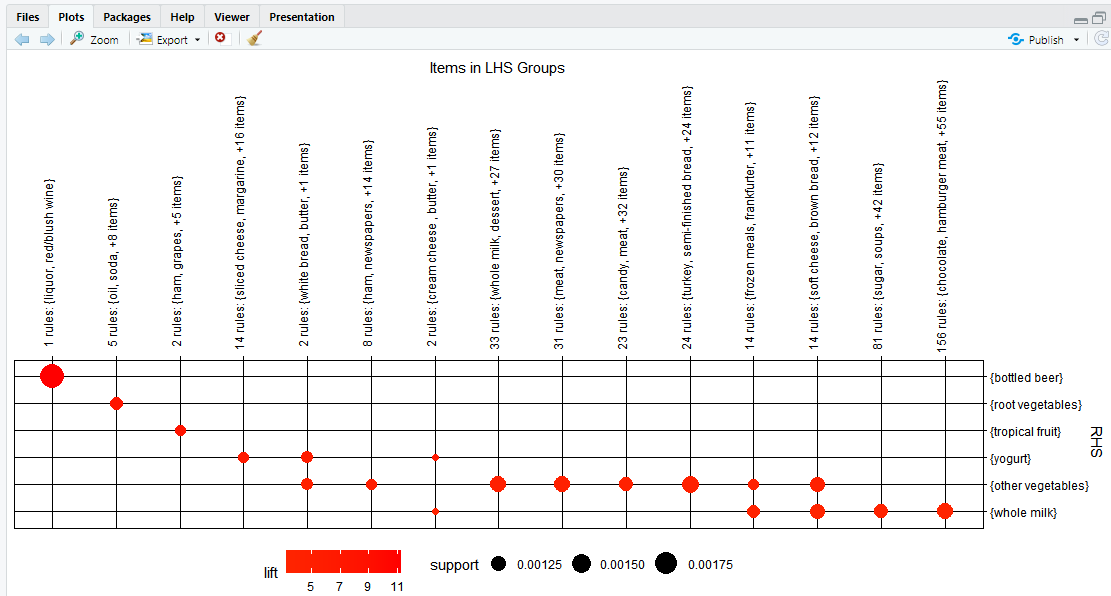
**Visualizaciones con la librería arulesViz**

****

****

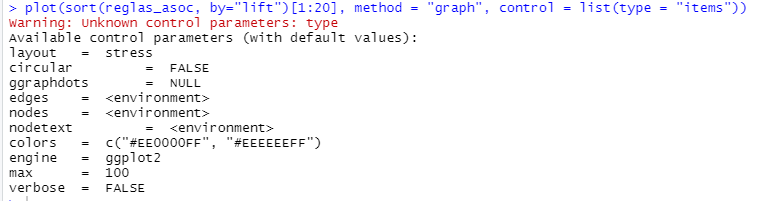
**En el eje horizontal están representados los miembros LHS (izquierda) de la regla y en el eje vertical los RHS (derecha). Cuanto mayor es el radio de la bola mayor es el soporte de la regla y cuanto más anaranjado el color mayor es el lift**

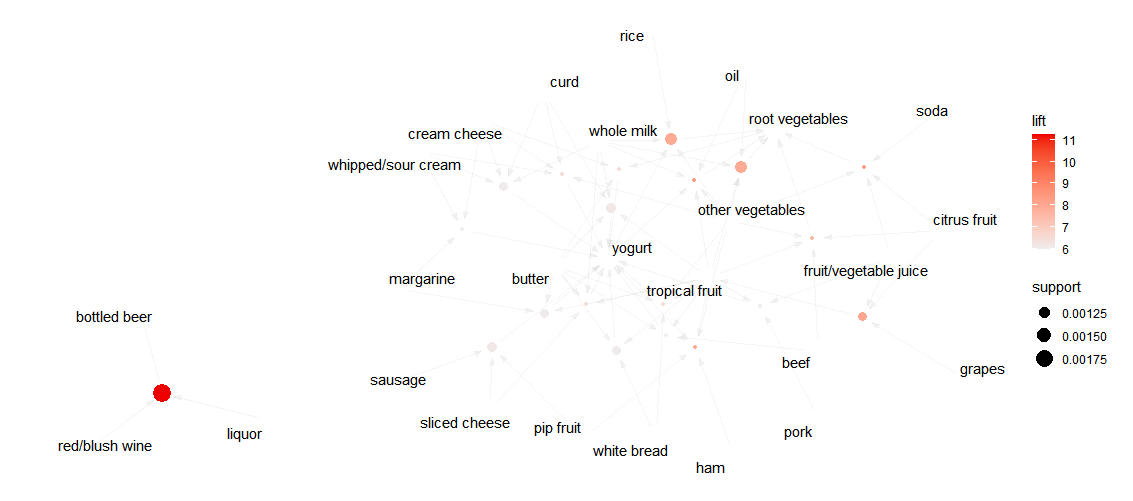
****

****

**Las reglas más importantes serían aquellas con un radio de bola mayor y que tuvieran un color más intenso**

**Representar las Reglas en modo grafo**

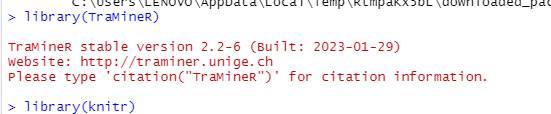
****

****

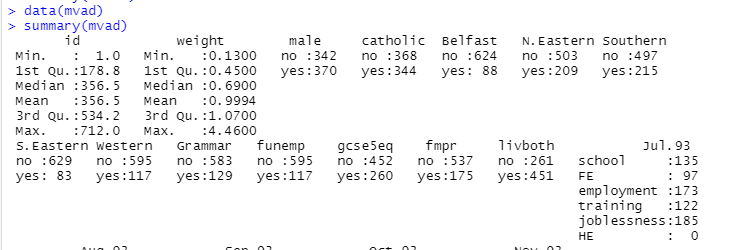
**B) Patrones Secuenciales. Ejemplo de aplicación de PSP con R**

Se cargan las librerias para el análisis de secuencias de estado o eventos; TraMineR

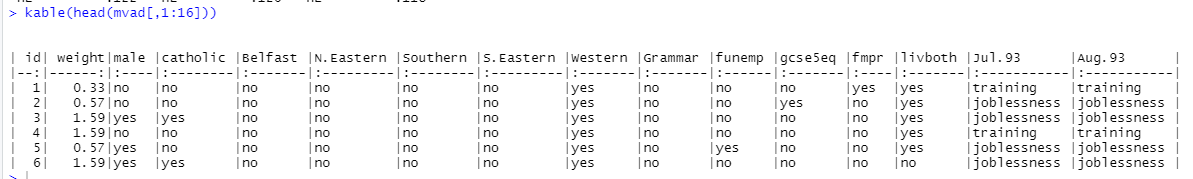


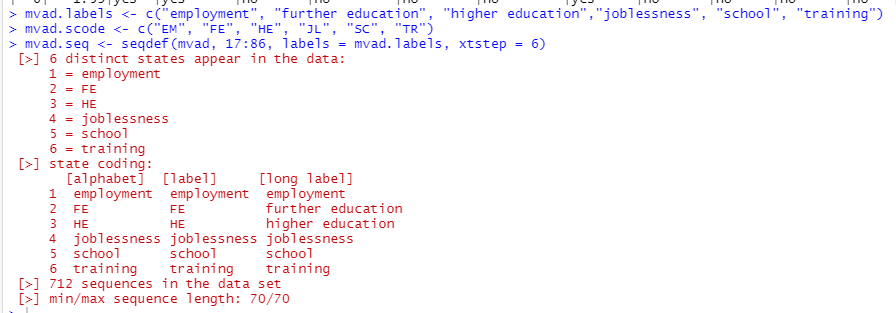


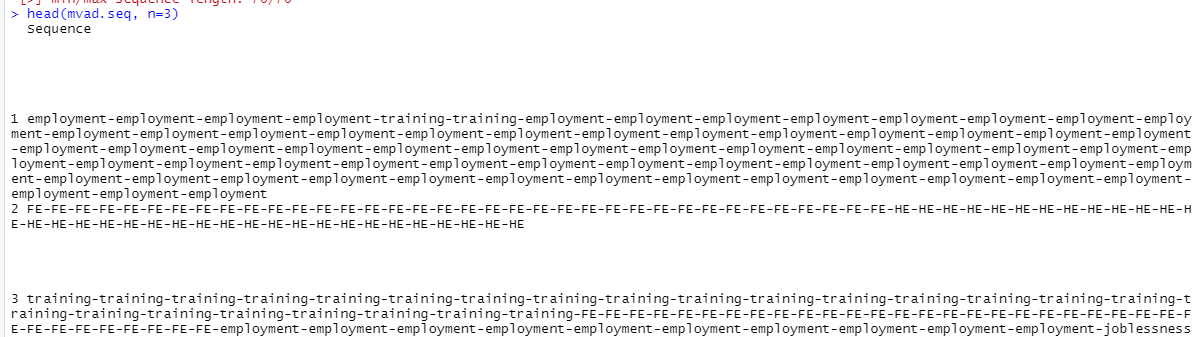
Partimos de un dataset (mvad) que está incluido en el paquete TraMineR y es un dataframe en el que se estudia la transición de estados desde la escuela al trabajo.

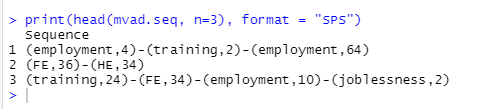


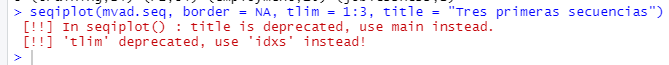
Los estados por los que puede pasar una persona según este dataset son: empleado (employment), educación superior (FE), educación universitaria (HE), desempleado (joblessness), en la escuela (school) y formándose (training)

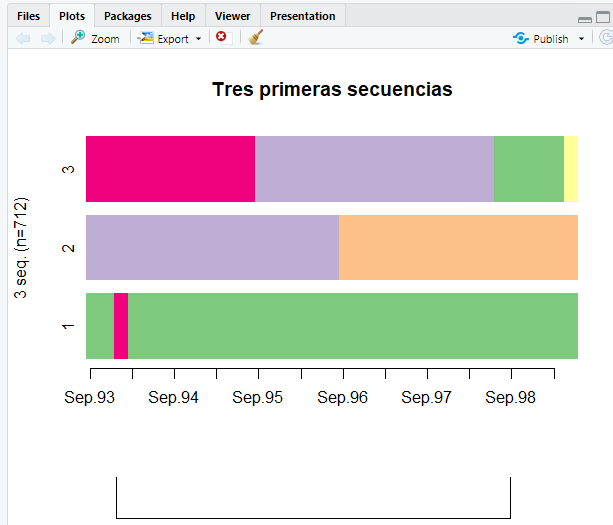




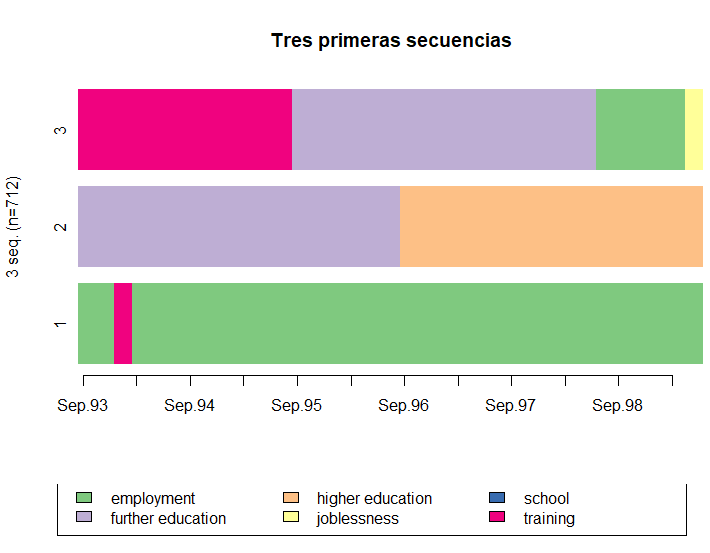


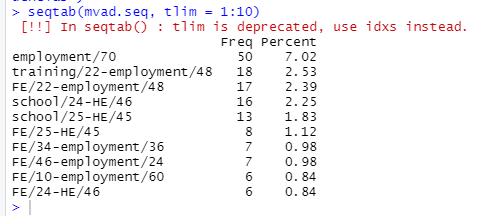


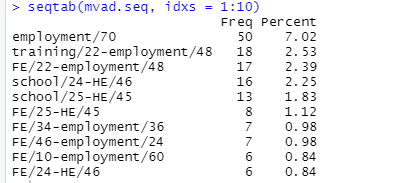




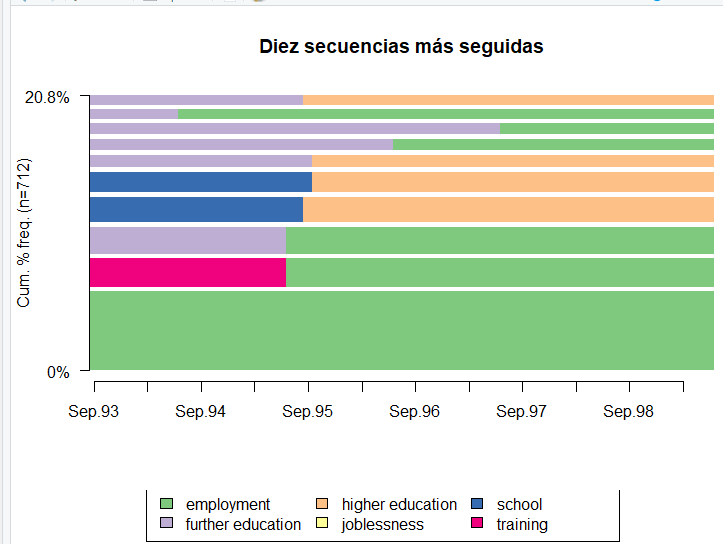










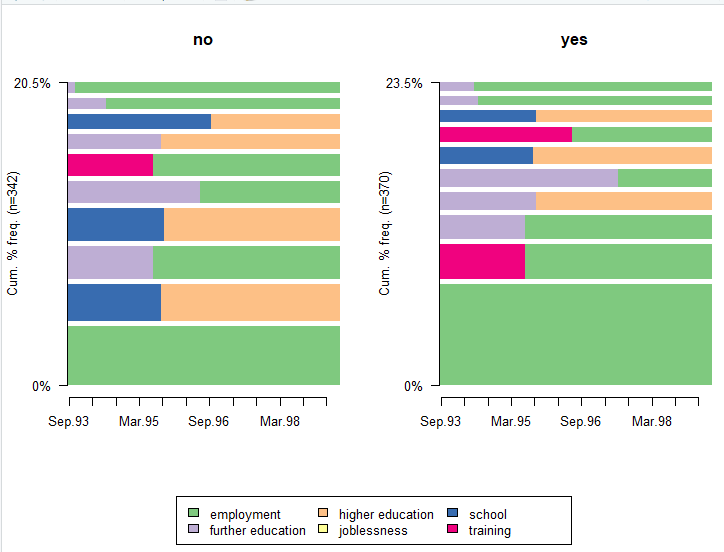


La secuencia de estados más frecuentes es la que corresponde a las personas que trabajan ininterrumpidamente en todo el periodo de estudio.

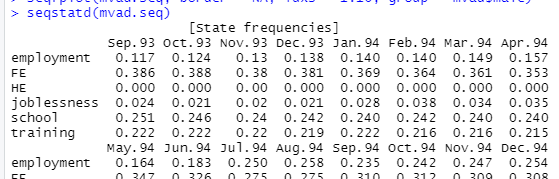
Calcular las tablas de secuencias frecuentes con una variable de agrupación

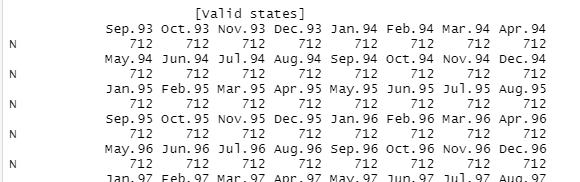
En este caso "male" del dataframe original mvad, es decir que podemos comparar las trayectorias más seguidas por hombres (male=1) y mujeres (male=0)

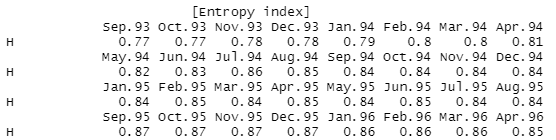




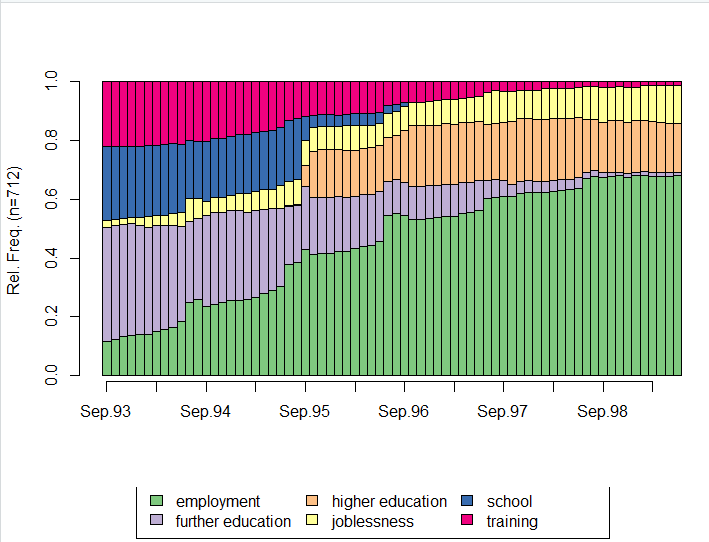
Con la función seqstatd() podemos obtener la tabla de distribución de estados por periodo (mes en este caso) y la gráfica correspondiente con la función seqdplot()







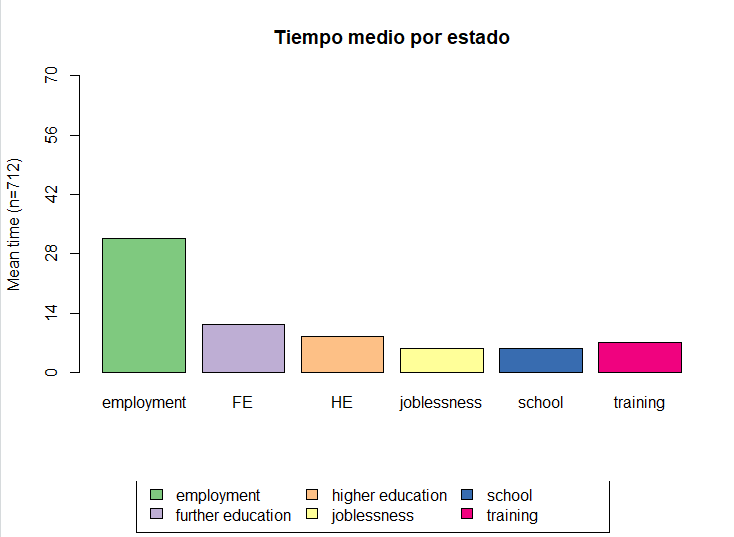




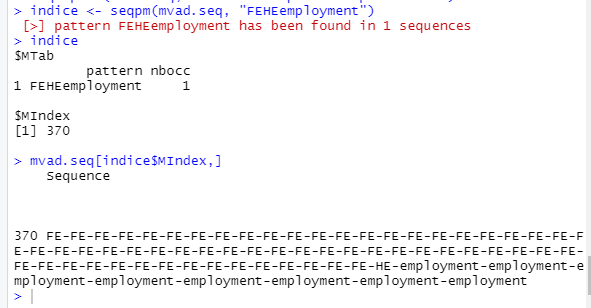
A medida que pasa el tiempo el estado más frecuente al final del periodo de estudio, como parecería obvio es que la persona esté trabajando (es casi el 60% de los estados de los usuarios en los últimos meses)

Se puede obtener el tiempo medio en cada estado utilizando la función seqmtplot():





Buscar las secuencias que cumpla algún patrón, por ejemplo, secuencias con FE, HE y employment



En este caso sólamente la secuencia 370 cumple con el criterio de búsqueda que hemos seleccionado

**C)Agrupación (clustering)**

El clustering es una actividad de clasificación no supervisada, pues no necesita que las observaciones estén etiquetadas como clases previamente.

Desde el punto de vista más matemático para fijar medidas de similitud, normalmente se utilizan conceptos como distancias, áreas de densidad, espacio de los datos, intervalos o distribuciones estadísticas.

Distancia Euclídea, Distancia de Manhattan, Distancia de Minkowsky, distancia ponderada

Variables numéricas. Euclídea o cualquier otra distancia de Minkowsky. Antes de aplicar la medida de la distancia conviene normalizar las variables para que las de un valor absoluto mayor no tengan demasiada influencia en dicha medida.

Variables binarias

Variables categóricas

Variables de distintos tipos mezcladas. Normalmente es lo que nos encontramos en los conjuntos de datos. Se recomienda usar una fórmula ponderada que combine las diferentes distancias normalizadas

**Medidas de calidad del clustering**

El objetivo del clustering es obtener una similitud entre los objetos que forman parte de un mismo cluster (intra-clase) muy alta y la similitud entre objetos de distintos clusters (inter-clase) muy baja. Cuanto más se cumpla esta máxima de mayor calidad será el clustering.

Dicha calidad depende en gran parte de la medida de similitud utilizada y el algoritmo de búsqueda utilizado

Se suele utilizar la suma de los cuadrados de las distancias entre observaciones de un mismo cluster (intra-clase) o de diferentes clusters (inter-clases)

**Modelos de clustering**

Algoritmos de particionado o partitioning algorithms (kmeans, pam, clara)

Algoritmos jerárquicos o Hierarchical algorithms

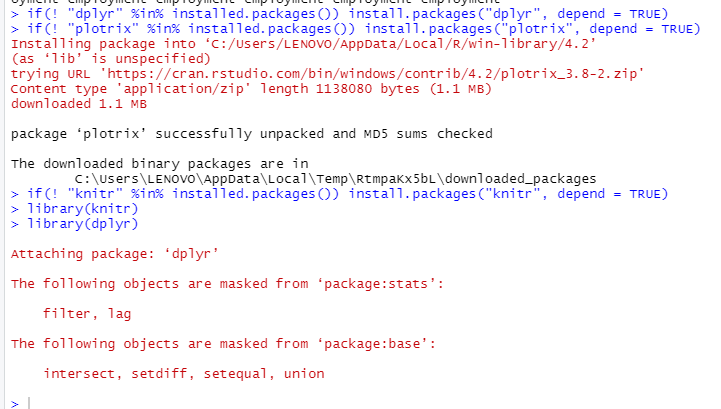
Algoritmos basados en densidad o Density-based algorithms

Algoritmos basados en maya o Grid-based algorithms

Algoritmos basados en modelo o Model-based algorithms

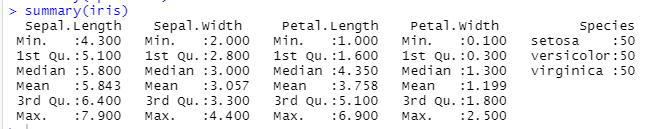
**Ejemplo de aplicación de K-Means con R**

Se cargan las librerías para el acondicionamiento de los datos y de representación gráfica

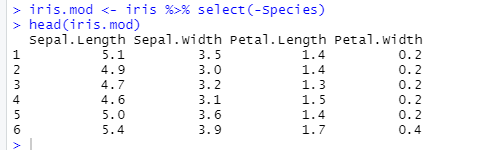
****

****

Utilizamos el famoso dataset de flores Iris incluido en R que contiene los datos de longitud y anchura del sépalo y el pétalo de 50 flores de las tres diferentes especies de iris: Setosa, Versicolor y Virgínica

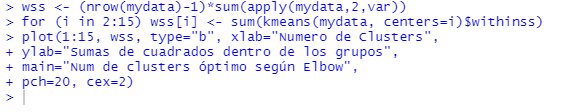


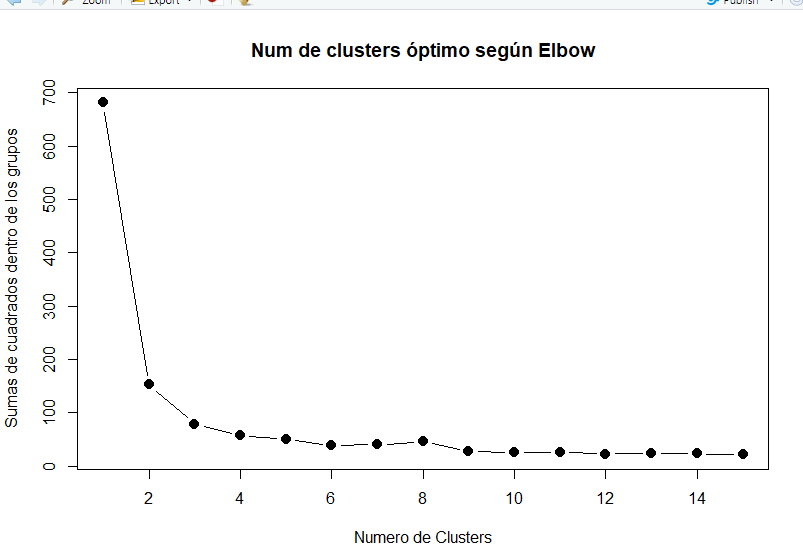
Le quitamos la variable Species que no queremos que se incluya en el clustering pues condicionaría mucho el resultado



Vamos a aplicar el clustering con K-means, pues todas son variables numéricas y podemos hacerlo. En este caso es necesario elegir el número de cluster final (k) antes de comenzar el análisis, pero el método de elbow nos puede dar una sugerencia del número ideal

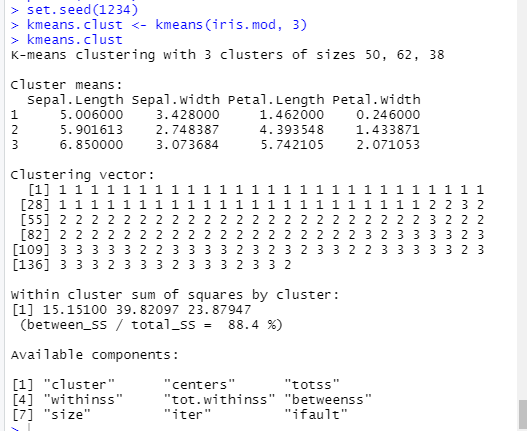






La gráfica intenta explicar la variación entre los clusters (suma de cuadrados). A partir del tercero la variación es muy pequeña, así que el número óptimo es k=3

Aplicamos entonces kmeans() con k=3



El resultado da gran información de salida en pantalla:

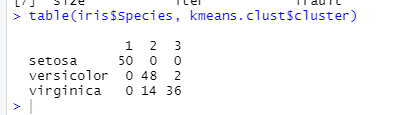
• El tamaño o número de elementos de cada cluster. Variable kmeans.clust.size

• Los centroides de cada cluster. Variable kmeans.clust.centers

• A que cluster corresponde cada observación en el dataset (se mantiene el orden del dataset inicial): Variable kmeans.clust.cluster

• El cuadrado de la suma de las distancias de todos los puntos al centroide de su cluster (WCSS). Variable kmeans.clust.withinss

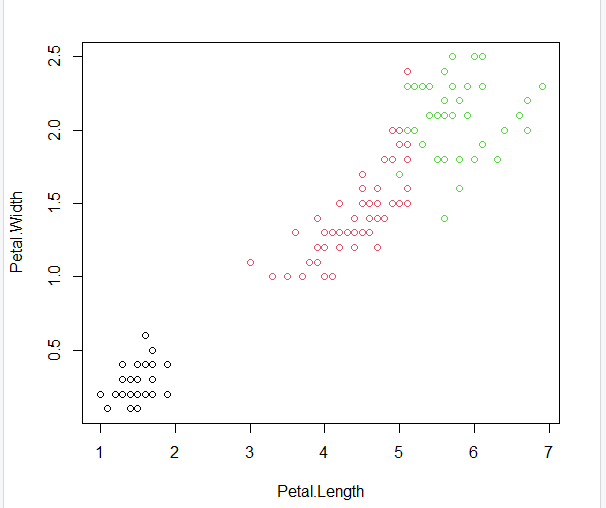
Vamos a comparar el resultado de la clusterización con la clasificación de las flores inicial que teníamos para ver si hay alguna relación.



Se ve bastante claramente que la variedad setosa corresponde con el cluster 1, la versicolor al cluster 2 y algo menos obvio, la virginica al cluster 3.

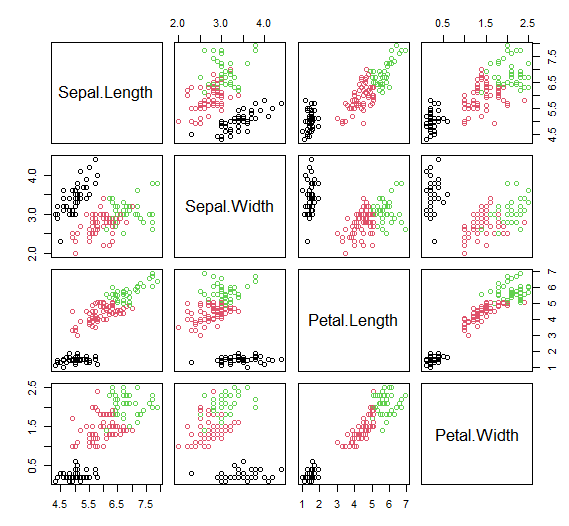
Si lo representamos gráficamente, sólo teniendo en cuenta las variables de los pétalos para que se vea mejor, percibiremos donde están los centroides y que en los clusters 2 y 3 se ve un poquito peor los contornos que los diferencian



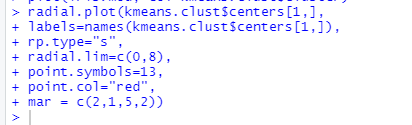


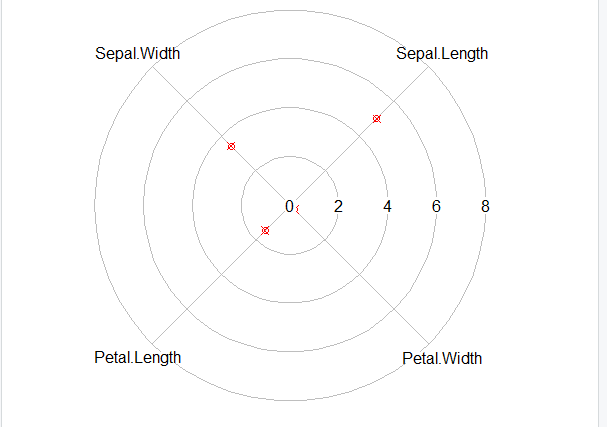
Incluso podemos representarlo confrontando las cuatro variables (las dos de los sépalos y las dos de los pétalos)



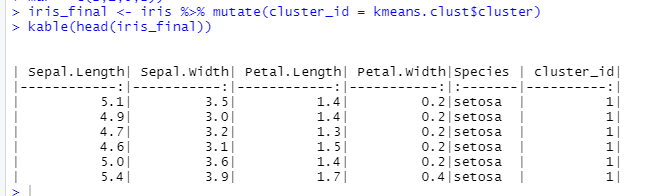


Una forma algo más elegante de representar los clusters es mediante un gráfico radial (radial.plot) o también conocido de gráfico de araña de la librería plotrix





Para obtener el resultado final del dataset original junto con el cluster al que se le ha asignado es muy sencillo utilizando la función mutate() de dplyr para crear una nueva columna que se hace corresponder con el valor de cluster para cada observación obtenido del vector de clusterizado



**Clustering jerárquico**

El cluster jerárquico (Hierarchical cluster analysis o HCA) es un método de clustering que busca construir una jerarquía de clusters.

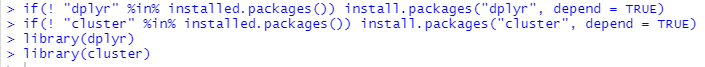
Tipos: Aglomerativo (bottom up) o Divisivo (top down). En general la complejidad del clustering aglomerativo es O(n3), lo que le hace bastante lento para datasets grandes. El clustering divisivo tiene incluso una complejidad peor O(2n).

Se suele representar en una gráfica llamada dendrograma.

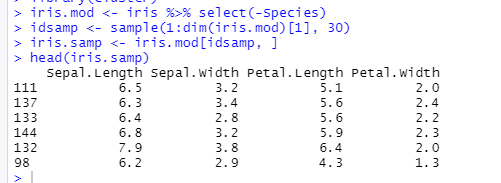
**Algoritmo agnes.** El algoritmo agnes es de tipo aglomerativo

**Implementación de hclust y agnes en R**

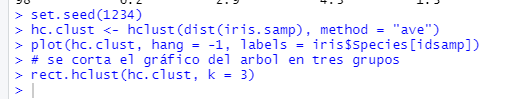
Se cargan las librerias para el clustering jerárquico y el acondicionamiento de datos

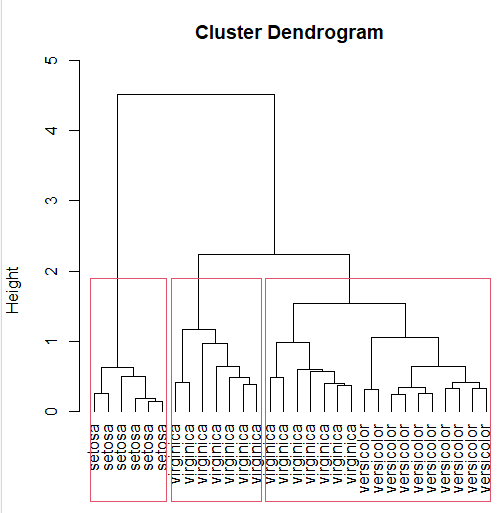


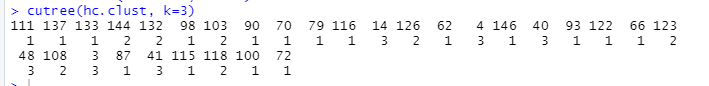
Quitamos la variable Species



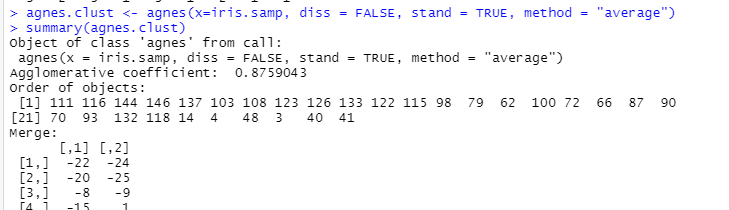
Aplicamos el método de clustering hclust() y representamos el dendrograma

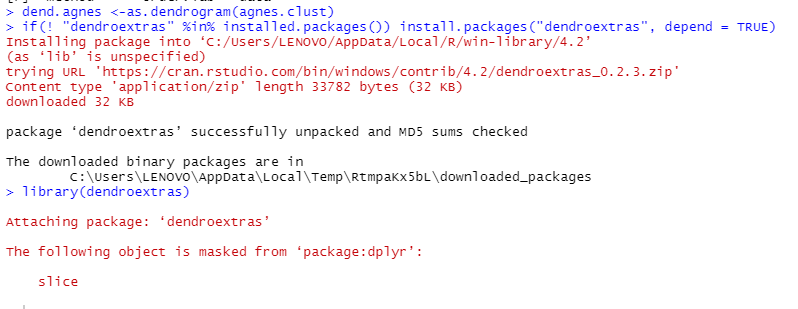




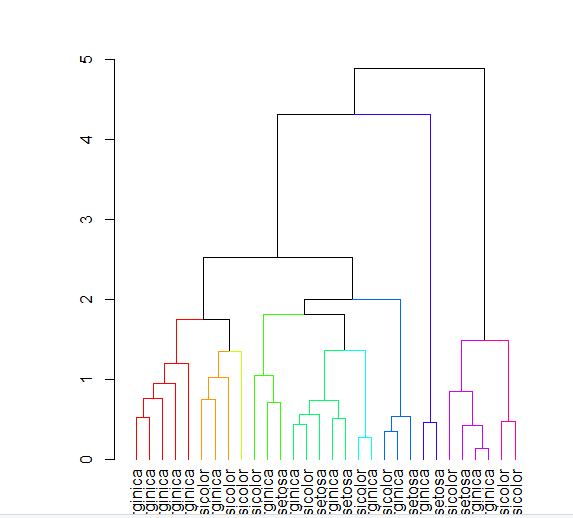


Ahora aplicamos el método de clustering agnes() del paquete cluster y representamos el dendrograma con la función plot() de la librería dendroextras para colorear los clusters. También podemos ver que la función summary() sobre el objeto resultado de aplicar agnes() da mucha información sobre el modelo.









Conclusión

Cualquiera de los modelos no supervisados que hemos visto:

• Reglas de Asociación

• Patrones Secuenciales

• Clustering

Ofrecen muchas posibilidades y se usan en multitud de casos de uso tanto para comprender los datos como para aplicarlos directamente para tareas de segmentación como incluso de predicción.

Tienen una **ventaja con respecto a los modelos supervisados** y es que no es necesario que las observaciones estén etiquetadas previamente, solamente acondicionar bien los datos y analizar.