```
---
title: "Métodos de agrupamiento usando Machine Learning"
autor: "Grupo 10"
format: html
---
```

GRUPO 10 - Integrantes

- ANNALISA CATERINA GUTIERREZ UTRILLA

- CARLOS RAMIRO HUARCAYA ANTEZANA

- EDWIN ADRIAN PECEROS ARENAS

- MILUSKA SARAI ZAMBRANO MOTTA

- YEZIT KATERIN QUISPE MONROY

Instalar y cargar los paquetes

```{r}
install.packages("factoextra")
install.packages("cluster")
```

```{r}
library(factoextra)
library(cluster)
library(here)
library(rio)
library(tidyverse)
```

```
Cargando paquete requerido: ggplot2
Welcome! Want to learn more? See two factoextra-related books at
https://goo.gl/ve3WBa
here() starts at D:/UPSJB_Practica_RStudio/estadistica_upsjb
Some optional R packages were not installed and therefore some file formats are
not supported. Check file support with show_unsupported_formats()
── Attaching core tidyverse packages
──────────────────────────────────────────────────────────────────────── tidyverse
2.0.0 ──
✓ dplyr     1.1.4     ✓ readr     2.1.5
✓ forcats   1.0.0     ✓ stringr   1.5.1
✓ lubridate 1.9.4     ✓ tibble    3.2.1
```

1 ¿Cómo aplicaremos Machine Learning a esta sesión?

Para intentar responder preguntas de investigación a veces es necesario que se realicen muchas medidas en una misma muestra. Por ejemplo, además de recolectar variables demográficas como la edad, genero y comobilidades, podríamos recolectar tambien varios otros parámetros adicionales. Y lo cierto es que es posible que existan patrones entre los valores de las variables. Es decir, es posible que haya una dependencia entre las variables predictoras. Por ejemplo, si un grupo de pacientes tienen insuficiencia renal aguda, algunos parámetros renales de laboratorio tendrán valores fuera del rango normal, mientras que otros parámetros, no. Un opción para aplicar técnicas convencionales es la excluir variables redundantes o variables que podamos encontrar como "no interesantes". No obstante, esto puede llevar a pérdida de información. Para estas situaciones se pueden usar técnicas de machine learning como las técnicas de agrupamiento (clustering), la cual permitan la inclusión de multiple variables y permite definir grupos de pacientes que comparten similitudes respecto a las variables incluídas.

VARIABLES DEMOGRÁFICAS: Edad, Genero, Fumador, Historia_Fumador, Historia_Radioterapia

VARIABLES CLÍNICAS: Funcion tiroidea, Examen fisico, Adenopatias

VARIABLES PATOLÓGICAS: Patologia, Focalidad, Riesgo

OTRAS VARIABLES

1.1 Uso de las técnicas de agrupamiento para responden preguntas de investigación en salud

Las técnicas de agrupamiento son un tipo de técnica exploratoria que puede usarse con el objetivo de clasificar observaciones (por ejemplo pacientes que forman parte de una muestra) en grupos en base a su similaridad y desimilaridad de las variables. A partir de esto, obtendremos grupos cuyos individuos que pertenecen a un mismo grupo son similares pero diferentes a individuos que pertenecen a otros grupos.

Los grupos encontrados pueden ser usados para hacer predicciones o evaluar diferencias en parámetros de laboratorio. Por ejemplo, entre grupos encontrados de pacientes quienes iniciaron su tratamiento para el cáncer, podemos comparar su supervivencia, calidad de vida luego de dos años u otras medidas a partir de los clusters (grupos) encontrados.

2 Análisis de agrupamiento herarquico (Hierarchical Clustering)

2.1 Sobre el problema para esta sesión

El conjunto de datos disponible contiene información de 383 pacientes diagnosticados con cáncer diferenciado de tiroides, incluyendo variables demográficas, clínicas, patológicas, de TNM y de seguimiento. Los datos incluyen variables numéricas (como la edad y el tamaño del tumor) y categóricas (como sexo, estado funcional tiroideo, tipo histológico, nivel de riesgo, estado de ganglios linfáticos y presencia de metástasis a distancia). El objetivo principal es aplicar el método de agrupamiento jerárquico para identificar grupos de pacientes con perfiles similares en sus variables clínicas y demográficas, con el fin de detectar patrones que puedan estar asociados con diferentes niveles de riesgo biológico, progresión o respuesta al tratamiento.

Este análisis facilitará la clasificación de pacientes en categorías que puedan tener implicaciones clínicas relevantes, ayudando a comprender mejor la heterogeneidad en los perfiles de la enfermedad y potencialmente a orientar decisiones terapéuticas y seguimiento en función de los patrones identificados en los grupos formados por el método de agrupamiento.

2.2 El dataset para esta sesión

Para ilustrar el proceso de análisis, utilizaremos el conjunto de datos llamado cáncer diferenciado de tiroides, que contiene datos de 383 pacientes con las siguientes variables: edad (años), sexo (masculino/femenino), tamaño del tumor (milímetros), tipo histológico (papilar/folicular), estado de ganglios linfáticos (presencia/ausencia de metástasis nodal), presencia de metástasis a distancia (sí/no), grado de diferenciación tumoral (grado I, II, III), nivel de riesgo biológico (bajo, intermedio, alto), respuesta al tratamiento (persistencia, remisión, progresión), y tiempo de seguimiento (meses). Este conjunto de datos se empleará para aplicar técnicas de agrupamiento jerárquico con el objetivo de identificar perfiles similares entre los pacientes, facilitando así la clasificación en grupos con características clínicas y pronósticas similares. Este enfoque permitirá detectar patrones que puedan estar asociados con diferentes niveles de riesgo y respuesta, aportando información relevante para la toma de decisiones clínicas y personalizadas en el manejo del cáncer de tiroides.

2.2.1 Importando los datos

```{r}
tiroides <- import(here("data", "tiroides.csv"))
```

2.3 Preparación de los datos

2.3.1 Solo datos numéricos

Para el análisis de agrupamiento jerárquico de esta sesión usaremos solo variables numéricas. Es posible emplear variables categóricas en esta técnica, pero esto no será cubierto aquí. El código abajo elimina las variables categóricas. `id` será el identificador para los participantes.

```{r}
tiroides_1 = tiroides |>
  select(-Genero, -Fumador, -Historia_Fumador, -Historia_Radioterapia,
-Funcion_Tiroidea, -Examen_Fisico, -Adenopatias, -Patologia, -T, -N, -M, -Riesgo,
-Respuesta, -Recurrencia, -Etapa, -Focalidad) |>
  column_to_rownames("Id")
```

2.3.2 La importancia de estandarizar

El documento destaca la importancia de estandarizar las variables antes de realizar un análisis de agrupamiento jerárquico en datasets clínicos, como el de cáncer de tiroides. La estandarización consiste en transformar todas las variables a una misma escala para asegurar que cada una tenga un peso comparable en el cálculo de las distancias entre objetos (en este caso, pacientes). Esto es fundamental porque las variables clínicas, como IMC o creatinina sérica, se miden en diferentes unidades y rangos numéricos, y sin estandarización, aquellas con valores mayores o diferentes unidades podrían dominar el análisis, generando agrupamientos sesgados.

Por ejemplo, si se consideran variables como IMC (en kg/m^2) y creatinina sérica (en mg/dL) sin estandarización, una diferencia de 1 en IMC (que puede ser de varias unidades) podría pesar igual que una diferencia de 1 en creatinina (que suele tener valores mucho menores), lo cual no sería correcto desde el punto de la relevancia clínica. Por ello, el uso de funciones como `scale()` en R, que transforma cada variable para que tenga media cero y desviación estándar uno, asegura que todas aporten de manera equitativa al cálculo de la distancia entre pacientes, facilitando una clasificación más precisa y representativa de la estructura real de los datos.

```{r}
tiroides_escalado = scale(tiroides_1)
```

Un vistazo a los datos antes del escalamiento:

```{r}
head(tiroides_1)
```

Description: df [6 × 1]

	Edad <int>
1	27
2	34
3	30
4	62
5	62
6	52

6 rows

y un vistazo después del escalamiento:

```{r}
head(tiroides_escalado)
```

```
        Edad
1 -0.9162408
2 -0.4537212
3 -0.7180181
4  1.3963572
5  1.3963572
6  0.7356149
```

2.4 Cálculo de distancias

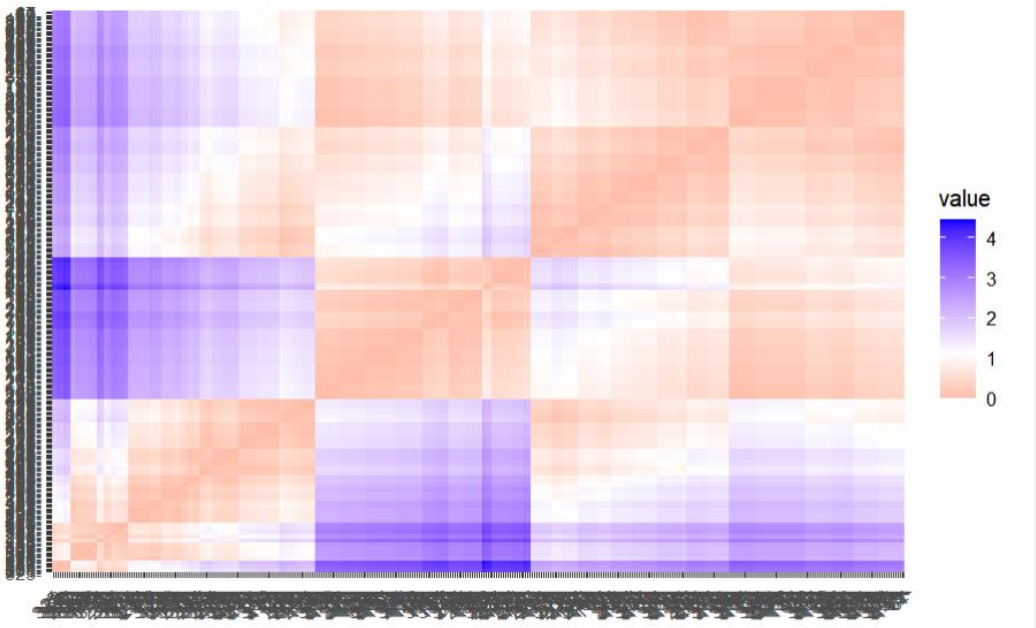
Dado que uno de los pasos es encontrar "cosas similares", necesitamos definir "similar" en términos de distancia. Esta distancia la calcularemos para cada par posible de objetos (participantes) en nuestro dataset. Por ejemplo, si tuvieramos a los pacientes A, B y C, las distancia se calcularían para A vs B; A vs C; y B vs C. En R, podemos utilizar la función `dist()` para calcular la distancia entre cada par de objetos en un conjunto de datos. El resultado de este cálculo se conoce como matriz de distancias o de disimilitud.

```{r}
dist_tiroides <- dist(tiroides_escalado, method = "euclidean")
```

2.4.1 (opcional) Visualizando las distancias euclidianas con un mapa de calor

Una forma de visualizar si existen patrones de agrupamiento es usando mapas de calor (heatmaps). En R usamos la función `fviz_dist()` del paquete factoextra para crear un mapa de calor.

```{r}
fviz_dist(dist_tiroides)
```



El nivel del color en este gráfico, es proporcional al valor de disimilaridad en observaciones (pacientes). Ejemplo, un color rojo puro indica una distancia con valor de 0 entre las abservaciones. Nota que la línea diagonal corresponde al intercepto de las mismas observaciones. Las observaciones que pertenecen a un mismo cluster (grupo) caen en orden consecutivo. Una conclusión del gráfico de abajo es que hay grupos que comparten similaridades dado que observamos grupos de colores.

2.5 El método de agrupamiento: función de enlace (linkage)

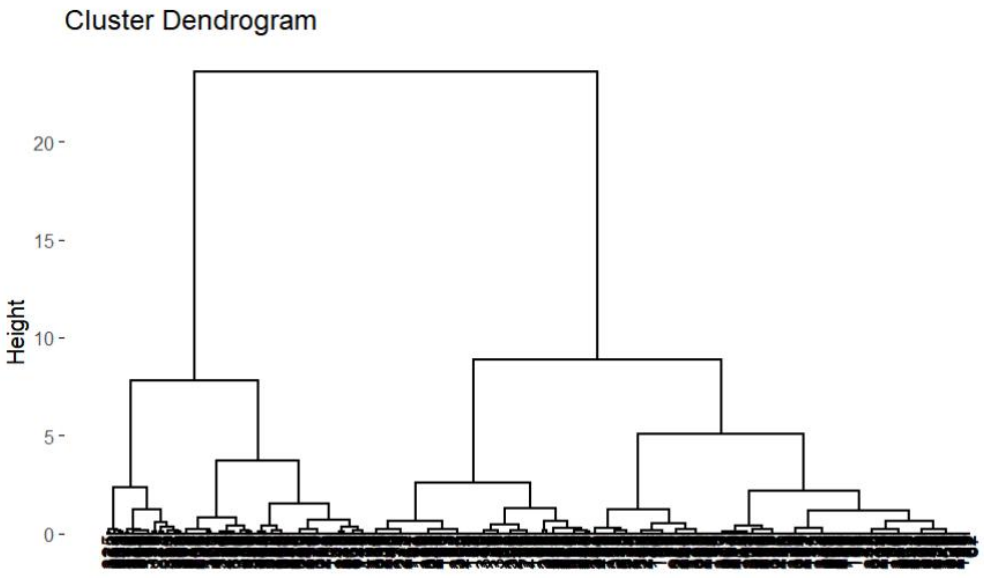
El agrupamiento jerárquico es un método que empieza agrupando las observaciones más parecidas entre sí, por lo que es fácil de usar al comienzo. Sin embargo, no basta con calcular las distancias entre todos los pares de objetos. Una vez que se forma un nuevo grupo (clúster), hay que decidir cómo medir la distancia entre ese grupo y los demás puntos o grupos ya existentes. Hay varias formas de hacerlo, y cada una genera un tipo diferente de agrupamiento jerárquico. La función de enlace (linkage) toma la información de distancias devuelta por la función `dist()` y agrupa pares de objetos en clústeres basándose en su similitud. Luego, estos nuevos clústeres formados se enlazan entre sí para crear clústeres más grandes. Este proceso se repite hasta que todos los objetos del conjunto de datos quedan agrupados en un único árbol jerárquico. Hay varios métodos para realizar este agrupamiento, incluyendo *Enlace máximo o completo*, *Enlace mínimo o simple*, *Enlace de la media o promedio*, *Enlace de centroide*, *Método de varianza mínima de Ward*. No entraremos en detalle sobre cómo funciona estos métodos, pero para este contexto el método de varianza minima de Ward o el método máximo, son preferidos. En este ejemplo, usamos el método de varianza mínima de Ward.

```r
{r}
dist_link_tiroides <- hclust(d = dist_tiroides, method = "ward.D2")
```

2.7 Dendrogramas para la visualización de patrones

Los dendrogramas es una representación gráfica del árbol jerárquico generado por la función `hclust()`.

```r
{r}
fviz_dend(dist_link_tiroides, cex = 0.7)
```

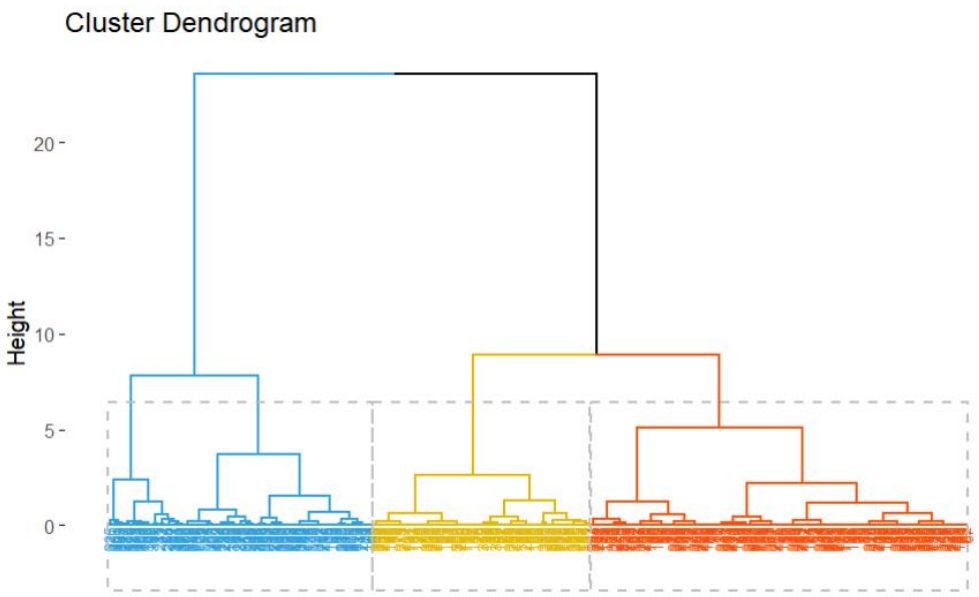


Un dendrograma es como un árbol genealógico para los clústeres (grupos). Esta muestra cómo los puntos de datos individuales o los grupos de datos se van uniendo entre sí. En la parte inferior, cada punto de datos se representa como un grupo independiente, y a medida que se asciende, los grupos similares se combinan. Cuanto más bajo es el punto de unión, mayor es la similitud entre los grupos.

2.8 ¿Cúantos grupos se formaron en el dendrograma?

Uno de los problemas con la agrupación jerárquica es que no nos dice cuántos grupos hay ni dónde cortar el dendrograma para formar grupos. Aquí entra en juego la decisión del investigador a partir de analizar el dendrograma. Para nuestro dendrograma, es claro que el dendrograma muestra tres grupos. En el código de abajo, el argumento k = 3 define el número de clusters.

```{r}
fviz_dend(dist_link_tiroides,
          k = 3,
          cex = 0.5,
          k_colors = c("#2E9FDF", "#E7B800", "#FC4E07"),
          color_labels_by_k = TRUE,
          rect = TRUE)
```



3 Agrupamiento con el algoritmo K-Means

El método de agrupamiento (usando el algoritmo) K-means es la técnica de machine learning más utilizado para dividir un conjunto de datos en un número determinado de k grupos (es decir, k clústeres), donde k representa el número de grupos predefinido por el investigador. Esto contrasta con la técnica anterior, dado que aquí sí iniciamos con un grupo pre-definido cuya idoniedad (de los grupos) puede ser evaluado. En detalle, el esta técnica clasifica a los objetos (participantes) del dataset en múltiples grupos, de manera que los objetos dentro de un mismo clúster sean lo más similares posible entre sí (alta similitud intragrupo), mientras que los objetos de diferentes clústeres sean lo más diferentes posible entre ellos (baja similitud intergrupo). En el agrupamiento k-means, cada clúster se representa por su centro (centroide), que corresponde al promedio de los puntos asignados a dicho clúster.

Aquí como funciona el algoritmo de K-Means

1. Indicar cuántos grupos (clústeres) se quieren formar. Por ejemplo, si se desea dividir a los pacientes en 3 grupos según sus características clínicas, entonces K=3.
2. Elegir aleatoriamente K casos del conjunto de datos como centros iniciales. Por ejemplo, R selecciona al azar 3 pacientes cuyas características servirán como punto de partida para definir los grupos.
3. Asignar cada paciente al grupo cuyo centro esté más cerca, usando la distancia euclidiana. Es como medir con una regla cuál centroide (paciente promedio) está más próximo a cada paciente en función de todas sus variables.
4. Calcular un nuevo centro para cada grupo. Es decir, calcular el promedio de todas las variables de los pacientes que quedaron en ese grupo. Por ejemplo, si en el grupo 1 quedaron 40 pacientes, el nuevo centroide será el promedio de la edad, genero, etc., de esos 40 pacientes. Este centroide es un conjunto de valores (uno por cada variable).
5. Repetir los pasos 3 y 4 hasta que los pacientes dejen de cambiar de grupo o hasta alcanzar un número máximo de repeticiones (en R, por defecto son 10 repeticiones). Esto permitirá que los grupos finales sean estables.

3.1 El problema y dataset para este ejercicio

Usaremos el mismo dataset, salvo que añadiremos otra variable numerica relacionado al orden de atención a los pacientes y el mismo problema que el que empleamos en el ejercicio anterior (para Agrupamiento Jerárquico).

```{r}
tiroidess1 <- import(here("data", "tiroides(1).csv"))
```

```{r}
tiroidess_1 = tiroidess1 |>
  select(-Genero, -Fumador, -Historia_Fumador, -Historia_Radioterapia,
-Funcion_Tiroidea, -Examen_Fisico, -Adenopatias, -Patologia, -T, -N, -M, -Riesgo,
-Respuesta, -Recurrencia, -Etapa, -Focalidad) |>
  column_to_rownames("Id")
```

3.2 Estimando el número óptimo de clusters

Como indiqué arriba, el método de agrupamiento k-means requiere que el usuario especifique el número de clústeres (grupos) a generar. Una pregunta fundamental es: ¿cómo elegir el número adecuado de clústeres esperados (k)?

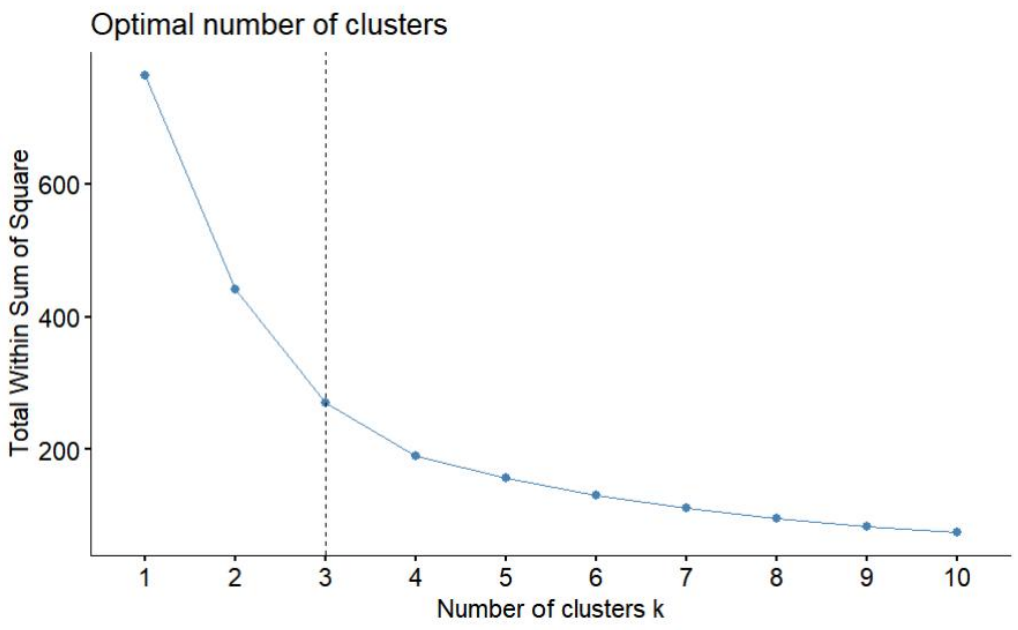
Aquí muestro una solución sencilla y popular: realizar el agrupamiento k-means probando diferentes valores de k (número de clústeres). Luego, se grafica la suma de cuadrados dentro de los clústeres (WSS) en función del número de clústeres. En R, podemos usar la función fviz_nbclust() para estimar el número óptimo de clústeres.

Primero escalamos los datos:

```{r}
tiroidess_escalado_2 = scale(tiroidess_1)
```

Ahora graficamos la suma de cuadrados dentro de los gráficos

```r
fviz_nbclust(tiroidess_escalado_2, kmeans, nstart = 25, method = "wss") +
  geom_vline(xintercept = 3, linetype = 2)
```



El punto donde la curva forma una "rodilla" o quiebre suele indicar el número óptimo de clústeres. Para nuestro gráfico, es en el número de cluster 3.

3.3 Cálculo del agrupamiento k-means

Dado que el resultado final del agrupamiento k-means es sensible a las asignaciones aleatorias iniciales se especifica el argumento `nstart = 25`. Esto significa que R intentará 25 asignaciones aleatorias diferentes y seleccionará la mejor solución, es decir, aquella con la menor variación dentro de los clústeres. El valor predeterminado de `nstart` en R es 1. Sin embargo, se recomienda ampliamente utilizar un valor alto, como 25 o 50, para obtener un resultado más estable y confiable. El valor emplead aquí, fue usado para determinar el número de clústeres óptimos.

```r
set.seed(123)
km_res <- kmeans(tiroidess_escalado_2, 3, nstart = 25)
```

```{r}
km_res
```

```
K-means clustering with 3 clusters of sizes 126, 163, 94

Cluster means:
        Edad      Orden
1 -0.5680401  0.7001189
2 -0.3673788 -0.9320478
3  1.3984659  0.6777533

Clustering vector:
  1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18  19  20
21  22  23  24  25  26  27  28  29  30  31  32
  2   2   2   2   2   2   2   2   2   2   3   2   2   2   3   2   2   2   2   2
2   2   2   3   2   2   2   2   2   2   2   2
 33  34  35  36  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52
53  54  55  56  57  58  59  60  61  62  63  64
  2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
2   2   2   2   2   2   2   2   2   2   2   2
 65  66  67  68  69  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84
85  86  87  88  89  90  91  92  93  94  95  96
  2   2   2   2   2   2   2   3   2   2   2   2   2   2   2   2   2   2   2   2
2   2   3   2   2   2   2   2   3   3   2   2   2   2
 97  98  99 100 101 102 103 104 105 106 107 108 109 110 111 112 113 114 115 116
117 118 119 120 121 122 123 124 125 126 127 128
  2   2   2   2   2   2   2   2   2   2   2   2   3   3   2   3   2   2   2   2
2   2   2   2   2   2   2   2   2   2   2   3
129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145 146 147 148
149 150 151 152 153 154 155 156 157 158 159 160
  3   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2   2
2   2   2   2   2   3   2   2   2   2   2   2
161 162 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
181 182 183 184 185 186 187 188 189 190 191 192
  2   2   2   2   2   2   2   3   1   2   2   1   2   2   3   2   2   3   2   3
3   2   2   1   3   3   1   1   3   1   1   1
193 194 195 196 197 198 199 200 201 202 203 204 205 206 207 208 209 210 211 212
213 214 215 216 217 218 219 220 221 222 223 224
  1   1   1   3   1   1   3   3   1   1   1   1   1   1   1   1   1   1   1   1
3   3   3   1   1   1   3   1   1   1   1   3
225 226 227 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244
245 246 247 248 249 250 251 252 253 254 255 256
  1   3   1   1   1   1   1   1   1   1   3   3   3   1   1   1   3   3   3   3
1   1   1   1   3   1   1   1   1   1   1   1
257 258 259 260 261 262 263 264 265 266 267 268 269 270 271 272 273 274 275 276
277 278 279 280 281 282 283 284 285 286 287 288
  1   1   1   3   1   3   1   1   1   3   1   1   1   1   1   1   1   1   1   1
3   1   3   1   1   1   3   1   1   3   3   3
289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308
309 310 311 312 313 314 315 316 317 318 319 320
  1   1   1   1   1   1   1   3   3   3   1   1   3   3   3   3   1   1   1   1
1   3   1   1   1   1   1   1   1   3   1   1
321 322 323 324 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340
 341 342 343 344 345 346 347 348 349 350 351 352
   3   3   3   3   3   3   1   3   1   3   3   3   1   1   1   1   3   1   1   3
 1   1   1   1   1   1   1   3   3   3   1   3
 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370 371 372
 373 374 375 376 377 378 379 380 381 382 383
   1   3   1   1   3   1   3   1   3   1   3   3   3   3   3   3   3   3   3   3
 3   1   3   3   1   1   3   3   3   3   3

 Within cluster sum of squares by cluster:
 [1]  60.72367 105.70325 104.54053
  (between_SS / total_SS =  64.5 %)

 Available components:

 [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
"betweenss"    "size"         "iter"
 [9] "ifault"
```

El resultado del análisis de agrupamiento (clustering) con K-means en el que se formaron 3 clústeres revela dos aspectos principales:

1. Medias o centros de los clústeres (Cluster means): Este es una matriz donde cada fila representa un clúster y cada columna una variable. En este caso, las variables son `Edad` y `Orden`. Los valores indican la posición media de cada variable dentro de cada clúster

   Clúster 1 tiene medias de aproximadamente -0.57 en Edad y 0.70 en Orden.

   Clúster 2 tiene medias de aproximadamente -0.37 en Edad y -0.93 en Orden.

   Clúster 3 tiene medias de aproximadamente 1.40 en Edad y 0.68 en Orden.

2. Vector de asignación de clúster (Clustering vector): Este vector indica a qué clúster pertenece cada uno de los 350 puntos (por ejemplo, pacientes) en función de su posición en las variables analizadas. Cada número en el vector (de 1 a 3) corresponde al clúster asignado:

   La mayoría de los puntos están en el clúster 2, seguido por puntos en el clúster 1 y el menos en el clúster 3.

Además, el análisis muestra la suma de cuadrados dentro de los clústeres, lo que indica la dispersión interna:

- Clúster 1 tiene la menor suma de cuadrados interna, sugiriendo menor variabilidad.
- Los clústeres 2 y 3 tienen sumas de cuadrados similares (aproximadamente 105 y 104), indicando mayor dispersión en estos grupos.

El porcentaje de variación explicada por la separación en estos clústeres es de aproximadamente 64.5%.

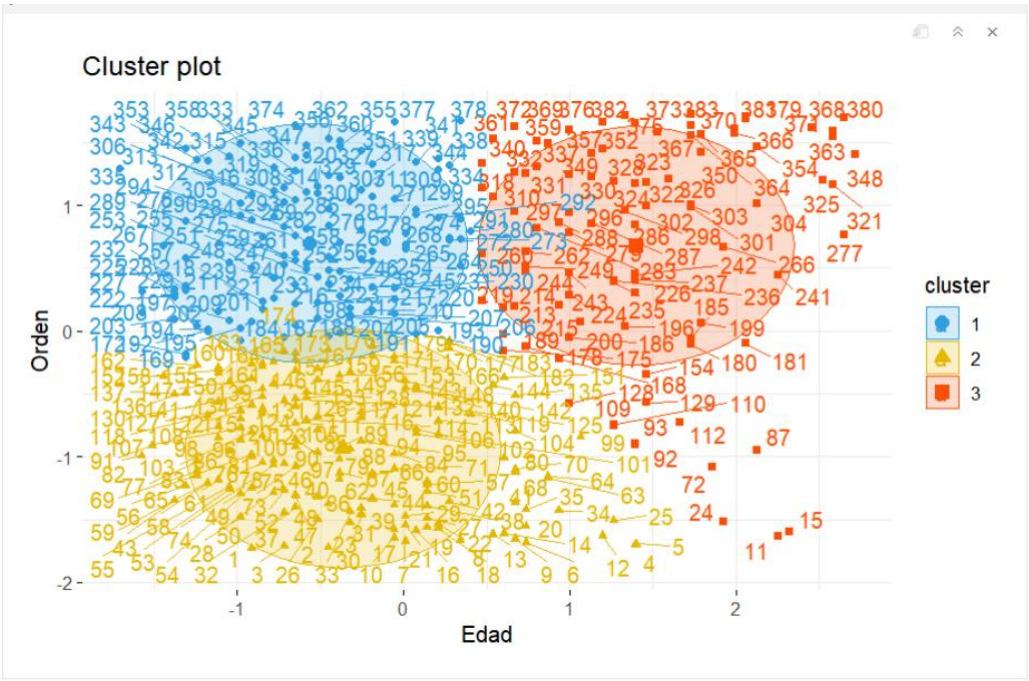
Este análisis permite entender la agrupación basada en las variables Edad y Orden, identificando grupos con características similares en estos aspectos

3.4 Visualización de los clústeres k-means

Al igual que el análisis anterior, los datos se pueden representar en un gráfico de dispersión, coloreando cada observación o paciente según el clúster al que pertenece. El problema es que los datos contienen más de dos variables, y surge la pregunta de qué variables elegir para representar en los ejes X e Y del gráfico. Una solución es reducir la cantidad de dimensiones aplicando un algoritmo de reducción de dimensiones, como el Análisis de Componentes Principales (PCA). El PCA transforma las 52 variables originales en dos nuevas variables (componentes principales) que pueden usarse para construir el gráfico.

La función `fviz_cluster()` del paquete factoextra se puede usar para visualizar los clústeres generados por k-means. Esta función toma como argumentos los resultados del k-means y los datos originales (hemo_data_escalado).

```{r}
fviz_cluster(
  km_res,
  data = tiroidess_escalado_2,
  palette = c("#2E9FDF", "#E7B800", "#FC4E07"),
  ellipse.type = "euclid",
  repel = TRUE,
  ggtheme = theme_minimal()
)
```



3.4.1 ¿Cómo interpretar?

En el gráfico resultante, los participantes (las observaciones) se representan como puntos. La técnica ha "creado" dimensiones, de las cuales dos de las más importantes de estas son consideradas en el gráfico. El uso aquí del PCA es poder clasificar diferentes "cosas" distintas en grupos, por ejemplo pacientes que fueron diagnosticados de cancer de tiroides y que tienen distintas edades, de una manera que genere el menor error posible. Además de los tres grupos formados (bien formados), nuestro gráfico aquí y en el agrupamiento jerárquico no nos dice más. Es necesario realizar análisis adicionales para evaluar la utilidad de estos clústeres.

Aviso sobre el dataset de esta sesión: A diferencia de sesiones anteriores, el conjunto de datos empleado en esta sesión es completamente simulado y no corresponde a información real de pacientes ni a datos provenientes de algún repositorio en línea. Es importante tener en cuenta que, en conjuntos de datos reales, los grupos formados mediante el análisis de agrupamiento pueden no ser tan claramente distinguibles como en estos ejemplos.