```
---
title: "Métodos de agrupamiento usando Machine Learning"
autor: "Grupo 01"
format: html
---
```

GRUPO 01 - Integrantes

- Bastidas Bendezu Ivan Luis
- Basurto Taype Marcelo Aaron
- Fonseca Moron José Kenneth
- Saravia Gutierrez Randy Esteban
- Talavera Ayllon Angel Ronaldo

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
Keep up to date with changes at https://tidyverse.org/blog/
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
✓ purrr     1.0.4     ✓ tidyr     1.3.1── Conflicts ────────────────────────────────────────
```

1 ¿Cómo aplicaremos Machine Learning a esta sesión?

Para intentar responder preguntas de investigación relacionadas con los datos recogidos en este estudio, es fundamental analizar múltiples variables simultáneamente. Por ejemplo, además de variables clínicas como la edad, el estadio tumoral, los Gleason y el volumen prostático, podemos incorporar otros parámetros, como el tiempo de almacenamiento de las unidades de glóbulos rojos, el volumen de sangre transfundido, el estado de antecedentes terapéuticos, y variables de seguimiento como la recurrencia bioquímica.

El uso de técnicas de Machine Learning permite detectar patrones complejos e interdependencias entre estas variables. Es decir, puede revelar dependencias entre los factores predictivos que, de manera individual, no sería fáciles de identificar. Por ejemplo, puede ocurrir que en pacientes con mayor riesgo de recurrencia bioquímica, existan combinaciones específicas de variables como el tiempo de almacenamiento de la sangre, el volumen de unidades transfundidas y el estadio T, que juntos contribuyen a un mayor riesgo, incluso si cada variable por separado no muestra una fuerte asociación.

De esta forma, los algoritmos de Machine Learning nos facilitarán modelar y predecir la recurrencia bioquímica, teniendo en cuenta las relaciones complejas entre todas estas variables, siempre con el objetivo de entender mejor los factores que influyen en los resultados clínicos.

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 dataset de esta sesión contiene información de 316 hombres que se sometieron a una prostatectomía radical, recibieron una transfusión durante o dentro de los 30 días posteriores al procedimiento quirúrgico y tenían datos de seguimiento de PSA disponibles. La principal exposición de interés fue el grupo de duración del almacenamiento de los glóbulos rojos (RBC). También se recopilaron varios factores demográficos, basales y pronósticos. El desenlace fue el tiempo hasta la recurrencia bioquímica del cáncer. El dataset incluye variables numéricas.

El objetivo principal de este análisis es utilizar el método de agrupamiento jerárquico para identificar grupos de pacientes que presenten características similares en sus perfiles clínicos y variables demográficas. Esto permitirá detectar patrones que puedan asociarse con diferentes grados de riesgo o progresión de la enfermedad, facilitando así la clasificación de pacientes en categorías con potenciales implicaciones clínicas.

2.2 El dataset para esta sesión

Para ilustrar el proceso de análisis, utilizaremos el dataset denominado `almac_sangr` que contiene información de 316 pacientes hombres sometidos a prostatectomía radical. Este conjunto de datos incluye las siguientes variables:

- **grupo_edad_GR**: Grupo de duración del almacenamiento de glóbulos rojos transfundidos, categorizado en "Joven" (≤13 días), "Intermedio" (13-18 días), y "Mayor" (≥18 días).
- **edad_mediana_GR**: Edad mediana de las unidades de glóbulos rojos transfundidas (días).
- **edad**: Edad del paciente en años.
- **raza_afroamericana**: Indicador si el paciente es de raza afroamericana ("Sí"/"No").
- **historia_familiar**: Presencia de historia familiar de la enfermedad ("Sí"/"No").
- **volumen_prostata**: Volumen de la próstata en gramos.
- **volumen_tumoral**: Categoría del volumen tumoral ("Bajo", "Medio", "Alto").
- **estadio_T**: Estadio clínico T del tumor ("T1-T2a" o "T2b-T3").
- **gleason_biopsia**: Puntuación Gleason en biopsia ("Gleason 0-6", "Gleason 7", "Gleason 8-10").
- **BN_positivo**: Estado del cuello vesical ("No"/"Sí").
- **confinamiento_organo**: Si el tumor está confinado al órgano ("No"/"Sí").

- **PSA_preoperatorio**: Nivel de antígeno prostático específico preoperatorio en ng/mL.
- **terapia_previa**: Recibió terapia previa ("No"/"Sí").
- **unidades_transfundidas**: Número de unidades transfundidas.
- **gleason_quirurgico**: Puntuación Gleason quirúrgica ("No asignado", "Gleason 0-6", "Gleason 7", "Gleason 8-10").
- **terapia_adyuvante**: Terapia adyuvante administrada ("No"/"Sí").
- **radioterapia_adyuvante**: Radioterapia adyuvante ("No"/"Sí").
- **recurrencia_bioquimica**: Evento de recurrencia bioquímica del cáncer de próstata ("No"/"Sí").
- **censura**: Criterio de censura en seguimiento ("No"/"Sí").
- **tiempo_hasta_recurrencia**: Tiempo hasta la recurrencia bioquímica en meses.

Estos datos permiten el análisis del impacto de diferentes variables clínicas y demográficas en el tiempo de recurrencia bioquímica tras la prostatectomía, incluyendo aspectos relacionados con la transfusión de glóbulos rojos y características tumorales de interés.

2.2.1 Importando los datos

```{r}
almac_sangr <- import(here("data", "almac_sangr.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.

```{r}
almac_sangr_1 = almac_sangr |>
  select(-Raza_afroamericana, -Grupo_edad_GR, -Historia_familiar, -Volumen_tumoral,
-Estadio_T, -Gleason_biopsia, -Gleason_quirurgico, -Confinamiento_organo,
-Terapia_previa, -Terapia_adyuvante, -Radioterapia_adyuvante,
-Recurrencia_bioquimica, -Censor, -BN_positivo) |>
  column_to_rownames("Id")
```

2.3.2 La importancia de estandarizar

Adicionalmente, es fundamental estandarizar las variables antes de realizar el análisis de agrupamiento jerárquico. Estandarizar implica transformar las variables para que compartan una escala común, facilitando su comparabilidad. Esto es especialmente importante porque, en el dataset, las variables clínicas se encuentran inicialmente medidas en diferentes unidades y rangos, lo cual puede influir de manera desproporcionada en el cálculo de las distancias entre los objetos (en este caso, los pacientes). Por ejemplo, la edad del paciente (en años), el volumen de la próstata (en gramos) y la puntuación Gleason en biopsia no se encuentran en la misma escala, por lo que comparar directamente sus valores puede inducir sesgos en los resultados del agrupamiento.

Para ilustrar: supongamos que consideramos variables como el volumen prostático y el nivel de PSA. El volumen se mide en gramos, mientras que el PSA en ng/mL. Sin una estandarización, una diferencia de 10 en volumen puede parecer significativa, mientras que en PSA puede ser menor o mayor en su impacto, pero no se puede determinar solo con los valores originales. La variable con mayor rango numérico o en diferente unidad podría dominar el cálculo de las distancias, sesgando los agrupamientos en favor de esa variable, sin reflejar necesariamente las verdaderas similitudes clínicas entre los pacientes.

Por ello, es recomendable aplicar funciones de estandarización, como la función `scale()` en R, que transforma las variables para que tengan media cero y desviación estándar uno. Esto asegura que todas las variables contribuyan de manera equitativa en la medición de las distancias, permitiendo que los grupos formados reflejen de manera más fiel las relaciones clínicas entre los pacientes, sin que ninguna variable predomine por su escala o rango original.

```{r}
almac_sangr_escalado = scale(almac_sangr_1)
```

Un vistazo a los datos antes del escalamiento:

```{r}
head(almac_sangr_1)
```

Description: df [6 × 6]

	Edad_mediana_GR <int>	Edad <dbl>	Volumen_prostata <dbl>	PSA_preoperato... <dbl>
1	25	72.1	54.0	14.08
2	25	73.6	43.2	10.50
3	25	67.5	102.7	6.98
4	15	65.8	46.0	4.40
5	15	63.2	60.0	21.40
6	25	65.4	45.9	5.10

6 rows | 1-5 of 6 columns

y un vistazo después del escalamiento:

```{r}
head(almac_sangr_escalado)
```

```
  Edad_mediana_GR      Edad Volumen_prostata PSA_preoperatorio
Unidades_transfundidas Tiempo_hasta_recurrencia
1       1.3200263 1.5113694      -0.08107066         0.9815876
1.8660551               -1.0591440
2       1.3200263 1.7185440      -0.43870486         0.3854605
-0.2399214                0.5151523
3       1.3200263 0.8760339       1.53159468        -0.2006757
-0.7664155               -0.6589170
4      -0.2720665 0.6412360      -0.34598488        -0.6302868
-0.2399214                0.9297357
5      -0.2720665 0.2821334       0.11761501         2.2004843
0.2865728               -1.1095663
6       1.3200263 0.5859894      -0.34929631        -0.5137256
-0.7664155                1.4630216
```

2.4 Cálculo de distancias

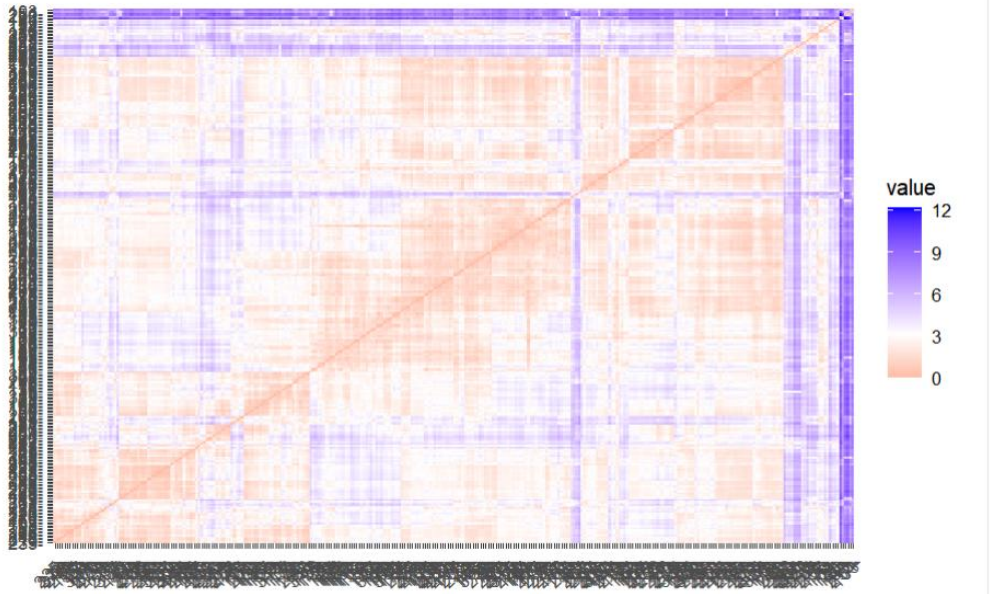
Dado que uno de los pasos es encontrar "cosas similares", necesitamos definir "similar" en términos de distancia. Esta distancia la calcularemos para cada par posible de objetos (participantes) en nuestro dataset. Por ejemplo, si tuvieramos a los pacientes A, B y C, las distancia se calcularían para 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_almac_sangr <- dist(almac_sangr_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_almac_sangr)
```



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)

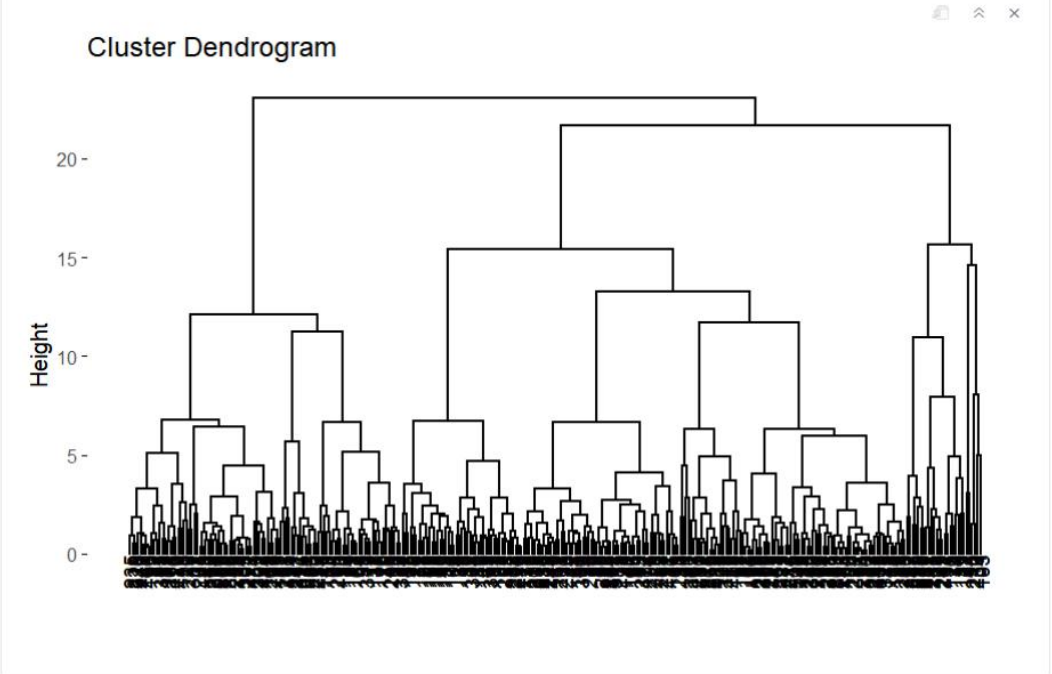
El agrupamiento jerárquico es un método que empieza agrupando las observaciones más parecidas entre sí, por lo que es fácil de usar al comienzo. Sin embargo, no basta con calcular las distancias entre todos los pares de objetos. Una vez que se forma un nuevo grupo (clúster), hay que decidir cómo medir 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}
dist_link_almac_sangr <- hclust(d = dist_almac_sangr, 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}
fviz_dend(dist_link_almac_sangr, cex = 0.7)
```

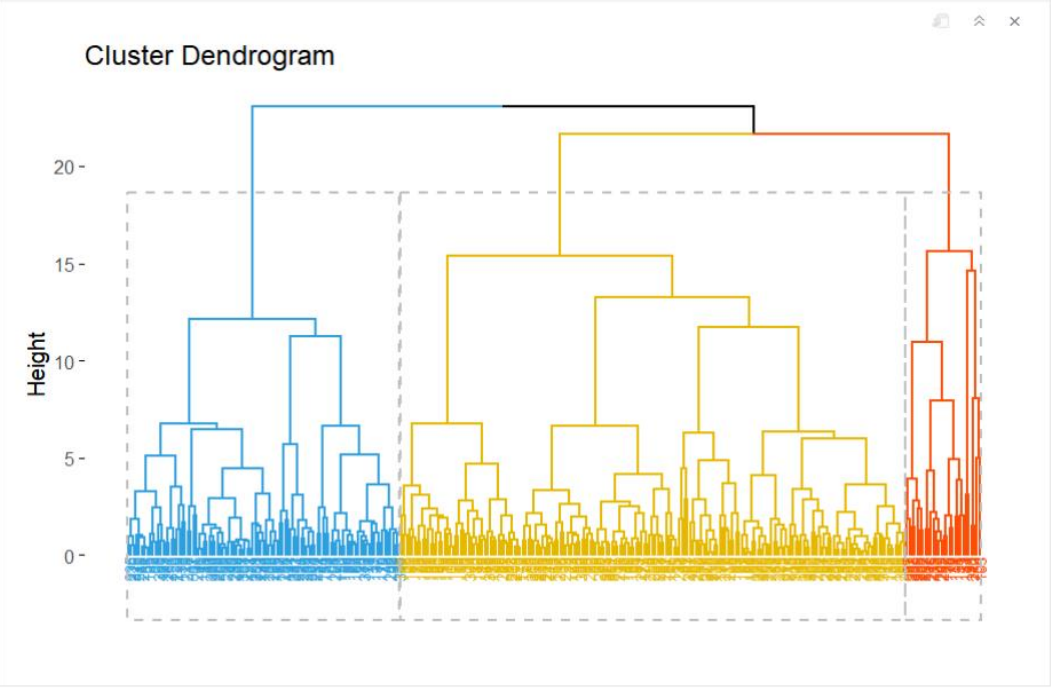


Un dendrograma es como un árbol genealógico para los clústeres (grupos). Esta muestra cómo los puntos de datos individuales o los grupos de datos se van uniendo entre sí. En la parte inferior, cada punto de datos se representa como un grupo independiente, y a medida que se asciende, los grupos similares 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_almac_sangr,
          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.
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 y el mismo problema que el que empleamos en el ejercicio anterior (para Agrupamiento Jerárquico).

```{r}
almac_sangr_2 = almac_sangr |>
  select(-Raza_afroamericana, -Grupo_edad_GR, -Historia_familiar, -Volumen_tumoral,
-Estadio_T, -Gleason_biopsia, -Gleason_quirurgico, -Confinamiento_organo,
-Terapia_previa, -Terapia_adyuvante, -Radioterapia_adyuvante,
-Recurrencia_bioquimica, -Censor, -BN_positivo, -Volumen_prostata,
-PSA_preoperatorio, -Tiempo_hasta_recurrencia) |>
  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)?

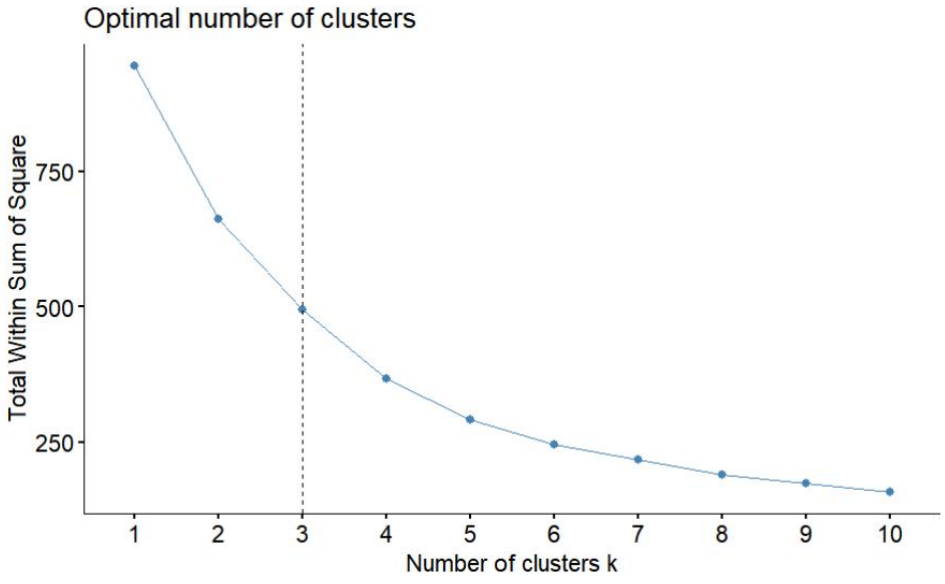
Aquí muestro una solución sencilla y popular: realizar el agrupamiento k-means probando diferentes valores de k (número de clústeres). Luego, se grafica la suma de cuadrados dentro de los clústeres (WSS) en función del número de clústeres. En R, podemos usar la función fviz_nbclust() para estimar el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

Primero escalamos los datos:

```{r}
almac_sangr_escalado_2 = scale(almac_sangr_2)
```

Ahora graficamos la suma de cuadrados dentro de los gráficos

```{r}
fviz_nbclust(almac_sangr_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 empleado aquí, fue usado para determinar el número de clústeres óptimos.

```{r}
set.seed(123)
km_res <- kmeans(almac_sangr_escalado_2, 3, nstart = 25)
```

```{r}
km_res
```

```
K-means clustering with 3 clusters of sizes 16, 100, 200

Cluster means:
  Edad_mediana_GR        Edad Unidades_transfundidas
1      0.07620381  0.11984658              3.2481022
2      1.32002628 -0.13193963             -0.2767760
3     -0.66610944  0.05638209             -0.1214602

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
  1   2   2   3   3   2   2   3   3   3   3   3   3   3   2   3   3   3   3   3
3   3   2   3   3   2   2   3   2   3   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   3   2   3   2   3   3   1   2   2   3   1   2   3   3   2   3   3   1   3
1   2   3   3   3   3   2   2   3   3   3   3
 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   3   3   2   3   2   3   2   2   3   1   3   2   3   3   3   1   2
3   3   3   3   3   2   3   3   2   3   3   3
 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
  3   1   3   3   3   3   1   2   3   3   3   2   3   3   2   3   3   3   3   3
3   3   2   2   2   3   3   2   3   3   3   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   3   3   3   2   3   3   2   2   2   2   3   2   3   3   2   2   2   3   3
3   2   2   2   3   2   3   3   3   3   3   3
```

```
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
  1   3   2   2   3   3   2   3   3   2   2   3   3   3   3   3   3   2   3   3
3   2   3   3   3   3   3   2   2   3   3   3
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   2   2   3   3   3   1   3   2   2   2   3   2   3   3   3   3   3   3   3
3   3   3   3   3   3   3   2   2   3   3   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
  3   2   3   3   2   3   2   3   3   2   2   3   3   3   2   2   2   2   3   3
3   3   3   3   3   3   3   3   3   2   3   2
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
  2   2   2   3   2   3   3   3   1   3   1   2   3   3   3   3   2   3   3   2
3   3   3   1   3   3   3   2   2   2   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
  2   3   2   2   3   3   3   3   3   2   3   3   2   3   2   3   3   3   1   2
3   3   3   3   2   2   3   2

Within cluster sum of squares by cluster:
[1] 100.9611 134.9200 264.0186
 (between_SS / total_SS =  47.1 %)

Available components:

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

El resultado del análisis de K-means con 3 clusters revela las siguientes características principales:

1. **Tamaños de los clusters**:

   Cluster 1: 16 observaciones

   Cluster 2: 100 observaciones

   Cluster 3: 200 observaciones

2. **Centroides de los clusters**:

- Cluster 1: mean de Edad Mediana_GR = 0.0762 y Edad Unidades_transfundidas = 0.1198
- Cluster 2: mean de Edad Mediana_GR = 1.3200 y Edad Unidades_transfundidas = -0.1319
- Cluster 3: mean de Edad Mediana_GR = -0.6661 y Edad Unidades_transfundidas = 0.0564

3. **Variabilidad dentro de los clusters (suma de cuadrados dentro)**:

- Cluster 1: 100.96
- Cluster 2: 134.92
- Cluster 3: 264.02

4. **Proporciones de variabilidad explicada (ratio entre la suma de cuadrados entre clusters y total)**: aproximadamente un 47.1%, indicando que los clusters contienen información relevante que separa las observaciones en grupos diferenciados.

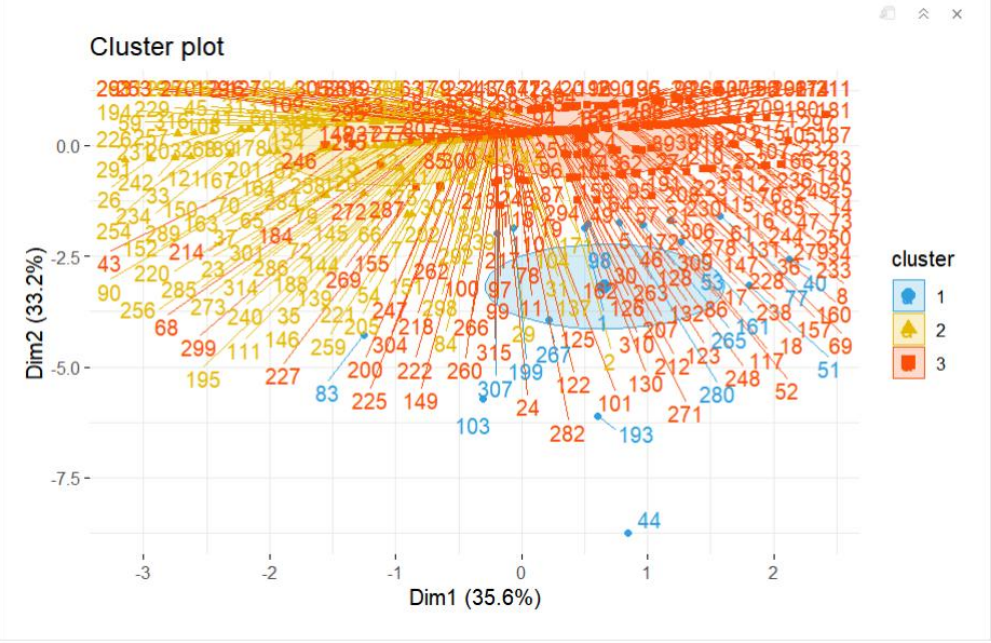
Esta información indica que los clusters difieren principalmente en las variables Edad Mediana_GR y Edad Unidades_transfundidas, lo cual puede responder a distintos patrones en los datos de acuerdo con estas características.

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).

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 = almac_sangr_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 iniciaron hemodialisis y que tienen distintos niveles de parámetros laboratoriales, de una manera que genere el menor error posible (en términos de predecir correctamente el tipo de célula). 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, como por ejemplo, evaluar si la supervivencia entre estos tres grupos varia. O evaluar como, en promedio, varían parámetros importantes de laboratorio.

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.