"Año de la recuperación y consolidación de la economía peruana"

UNIVERSIDAD PRIVADA SAN JUAN BAUTISTA

FACULTAD DE CIENCIAS DE LA SALUD

ESCUELA PROFESIONAL DE MEDICINA HUMANA

CURSO:

SISTEMATIZACIÓN Y MÉTODOS ESTADÍSTICOS

DOCENTE:

SEGUNDO VICENTE CASTRO LOPEZ

INTEGRANTES:

- **Marroquin Ticona Dayana**
- **Degregori Hinojosa Alexandra**
- **Loayza Quispe Leidy**
- **Quispe Valencia Evelyn**
- **Cayllahua Huarancca Mayrha**

```
---
title: "SEMANA 13 G1"
format: html
editor: visual
---
```

Integrantes

Dayana Lilian Marroquín Ticona

Alexandra Pamela Degregori Hinojosa

Cayllahua Huarancca Mayrha

Leidy Yojhana Loayza Quispe.

Evelyn Lizbeth Quispe Valencia

Instalar (si es necesario)

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

Cargar paquetes

```{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 C:/Users/Oficina/Desktop/GRUPO 1/G1
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
──────────────────────────────────────────────────────────────────────────────
──────────────────────────────────────────────────── tidyverse_conflicts() ──
✖ dplyr::filter() masks stats::filter()
✖ dplyr::lag()    masks stats::lag()
ℹ Use the conflicted package to force all conflicts to become errors
```

2 Análisis de agrupamiento herarquico

2.1 Sobre el problema para esta sesión

El dataset de esta sesión contiene información clínica de pacientes diagnosticados con cáncer de próstata y tratados en un contexto hospitalario. Este conjunto de datos incluye variables clínicas, anatómicas y de laboratorio, como la edad, el volumen prostático, el nivel de PSA preoperatorio, el puntaje de Gleason, y otras variables relacionadas con tratamientos previos, recurrencia bioquímica y resultados oncológicos a mediano plazo.

El objetivo de este ejercicio es aplicar el método de agrupamiento jerárquico para identificar grupos de pacientes con características similares en cuanto a su estado clínico basal y respuesta al tratamiento. Esto permitirá proponer posibles perfiles de riesgo, patrones clínicos diferenciados, o subgrupos que podrían beneficiarse de estrategias terapéuticas específicas.

2.2 El dataset para esta sesión

El dataset `almac_sangre` contiene información clínica de pacientes con cáncer de próstata, incluyendo variables demográficas, anatómicas, clínicas y de tratamiento. Entre ellas se encuentran la edad, raza, antecedentes familiares, volumen prostático, estadio clínico, PSA preoperatorio y puntuaciones de Gleason. Además, se consideran datos sobre tratamientos recibidos, recurrencia bioquímica y desenlaces postoperatorios. Este conjunto de variables permite aplicar técnicas de agrupamiento para identificar perfiles clínicos diferenciados y posibles patrones de evolución.

2.2.1 Importando los datos

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

2.3 Preparación de los datos

2.3.1 Creacion del ID

```{r}
sangre_data <- sangre_data |>
  mutate(id = paste0("Paciente_", str_pad(row_number(), 3, pad = "0")))
```

2.3.2 Solo datos numéricos

Para el análisis de agrupamiento jerárquico de esta sesión utilizaremos únicamente variables numéricas. Si bien es posible incluir variables categóricas mediante transformaciones como codificación one-hot, esta versión del análisis se centrará solo en datos continuos para facilitar la interpretación. En el código siguiente se eliminan las variables categóricas como Grupo_edad_GR, Raza_afroamericana, Historia_familiar, Volumen_tumoral, Estadio_T, Gleason_biopsia, Confinamiento_organo, Terapia_previa, Gleason_quirurgico, Terapia_adyuvante, Radioterapia_adyuvante, Recurrencia_bioquimica, Censor y BN_positivo.

```{r}
sangre_data_1 <- sangre_data |>
  select(
    id,
    Edad,
    Edad_mediana_GR,
    Volumen_prostata,
    PSA_preoperatorio,
    Unidades_transfundidas,
    Tiempo_hasta_recurrencia
  ) |>
  drop_na() |>
  column_to_rownames("id")
```

2.3.3 La importancia de estandarizar

Dado que las variables numéricas de este dataset, como el PSA preoperatorio, el volumen prostático y las unidades transfundidas, se expresan en escalas diferentes, es necesario estandarizarlas antes de aplicar el análisis de agrupamiento. Esto evita que variables con valores absolutos más grandes dominen el cálculo de distancias y distorsionen la formación de grupos.

La estandarización transforma las variables para que tengan media cero y desviación estándar uno, asegurando que todas contribuyan equitativamente al análisis. En R, esto se realiza fácilmente con la función scale().

Vistazo del escalamiento:

```r
head(sangre_data_escalado)
```

```
                  Edad Edad_mediana_GR Volumen_prostata PSA_preoperatorio
Unidades_transfundidas Tiempo_hasta_recurrencia
Paciente_001 1.5313837       1.3203302       -0.0826847         0.9559503
1.8536442               -1.0622417
Paciente_002 1.7385178       1.3203302       -0.4387581         0.3655701
-0.2314912                0.5030770
Paciente_003 0.8961727       1.3203302        1.5229428        -0.2149155
-0.7527750               -0.6642970
Paciente_004 0.6614208      -0.2630244       -0.3464428        -0.6403850
-0.2314912                0.9152961
Paciente_005 0.3023884      -0.2630244        0.1151339         2.1630964
0.2897927                -1.1123765
Paciente_006 0.6061850       1.3203302       -0.3497398        -0.5249476
-0.7527750                1.4455409
```

Interpretación:

Cada valor representa cuántas desviaciones estándar se aleja ese dato del promedio general de su variable:

Valores positivos indican que ese paciente tiene un valor mayor al promedio en esa variable.

Valores negativos indican que el paciente está por debajo del promedio en esa medida.

Paciente_001 tiene: Edad mayor al promedio (+1.53 desviaciones estándar), PSA preoperatorio también elevado (+0.96), Tiempo hasta recurrencia muy por debajo del promedio (-1.06), lo cual podría sugerir una recurrencia temprana.

Paciente_003 muestra un volumen prostático mucho mayor que el promedio (+1.52), pero una edad cercana al promedio y pocas unidades transfundidas.

2.4 Cálculo de distancias

Uno de los pasos clave en el análisis de agrupamiento es establecer qué tan similares o diferentes son los pacientes entre sí. Para ello, utilizamos una medida de distancia que se calcula entre cada par de pacientes en función de sus variables clínicas numéricas estandarizadas, como la edad, volumen prostático, PSA y tiempo hasta recurrencia.

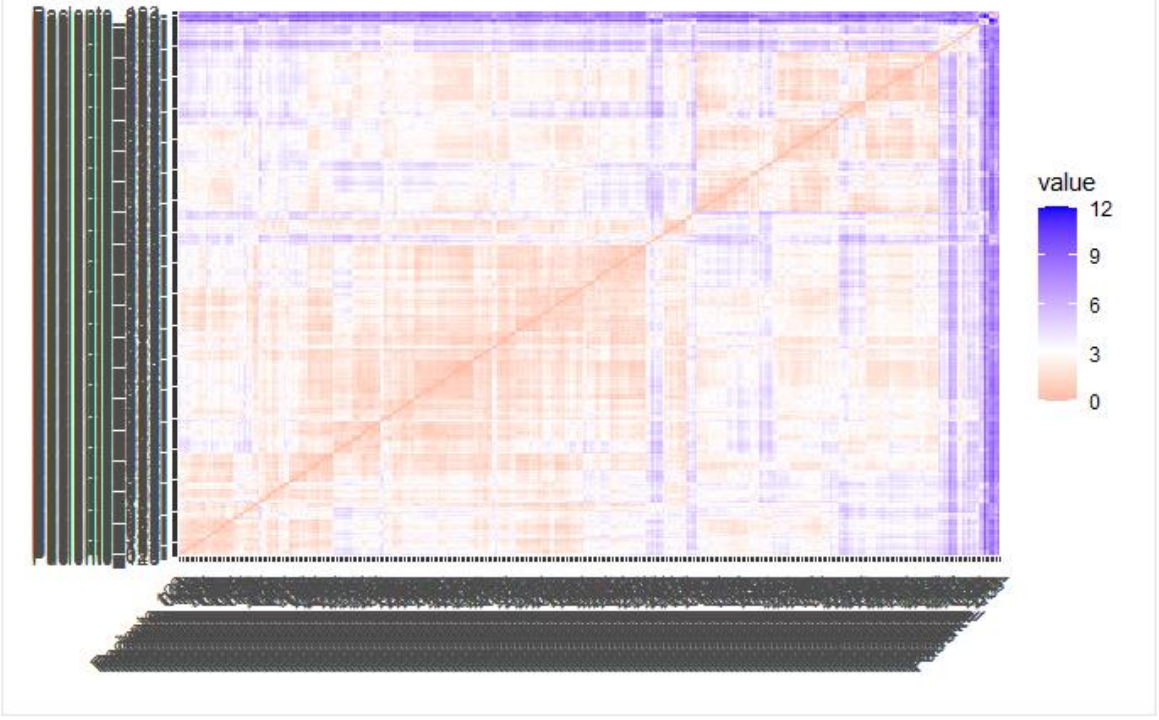
En este caso, trabajamos con la versión escalada del dataset (`sangre_data_escalado`), lo que garantiza que todas las variables estén en la misma escala. Utilizaremos la función `dist()` en R para calcular la distancia euclidiana entre cada par de pacientes. El resultado será una matriz de distancias, también llamada matriz de disimilitud, que servirá como base para construir el dendrograma y realizar el agrupamiento.

```r
dist_sangre_data <- dist(sangre_data_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_sangre_data)
```



Interpretación:

Cada celda del mapa representa la distancia euclidiana entre dos pacientes según sus variables clínicas estandarizadas (edad, volumen prostático, PSA, etc.).

Celdas en colores más oscuros (azul/violeta) indican que los pacientes están más alejados entre sí en el espacio de variables, es decir, son muy distintos.

Celdas en colores claros o blancos indican que los pacientes tienen valores similares en las variables analizadas, es decir, son clínicamente parecidos.

La diagonal principal siempre es cero (mismo paciente comparado consigo mismo) y aparece de color claro o blanco total.

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

En este análisis, utilizamos el método de agrupamiento jerárquico para identificar grupos de pacientes con características clínicas similares. Este método parte de una matriz de distancias (calculada previamente) y va uniendo de manera progresiva los pacientes o grupos más cercanos entre sí.

Una vez agrupados los pacientes iniciales, se debe decidir cómo calcular la distancia entre un grupo y los demás. Esto se realiza mediante una función de enlace (linkage), que determina cómo medir la similitud entre grupos a medida que se van formando clústeres más grandes.

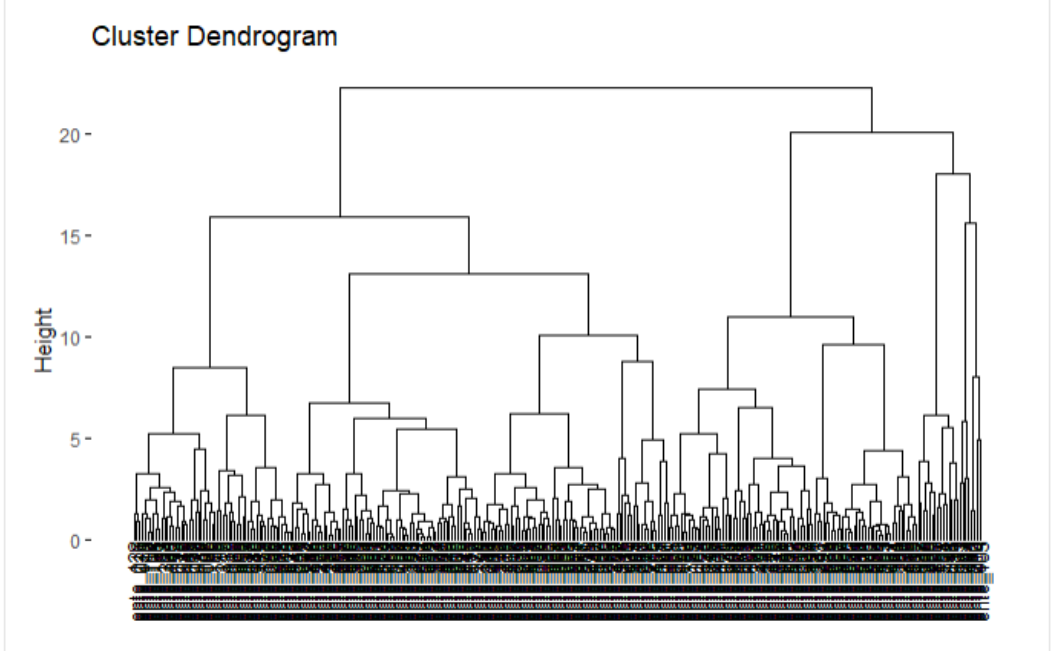
Existen diversos métodos de enlace, como el enlace completo, el mínimo, el promedio o el de centroides, entre otros. En este caso, utilizamos el método de varianza mínima de Ward, que busca minimizar la variabilidad dentro de cada grupo en cada paso del proceso. Este método es especialmente útil cuando se desea obtener grupos homogéneos y bien diferenciados en función de múltiples variables clínicas estandarizadas.

```{r}
dist_link_sangre_data <- hclust(d = dist_sangre_data, 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_sangre_data, cex = 0.7)
```



Interpretación:

El gráfico que observas es un dendrograma, una representación visual del proceso de agrupamiento jerárquico que muestra cómo los pacientes se agrupan progresivamente en función de su similitud clínica.

El eje vertical indica la distancia o disimilitud entre los grupos cuando se fusionan. Cuanto más alto es el punto de unión, más diferentes eran los grupos que se unieron en ese paso.

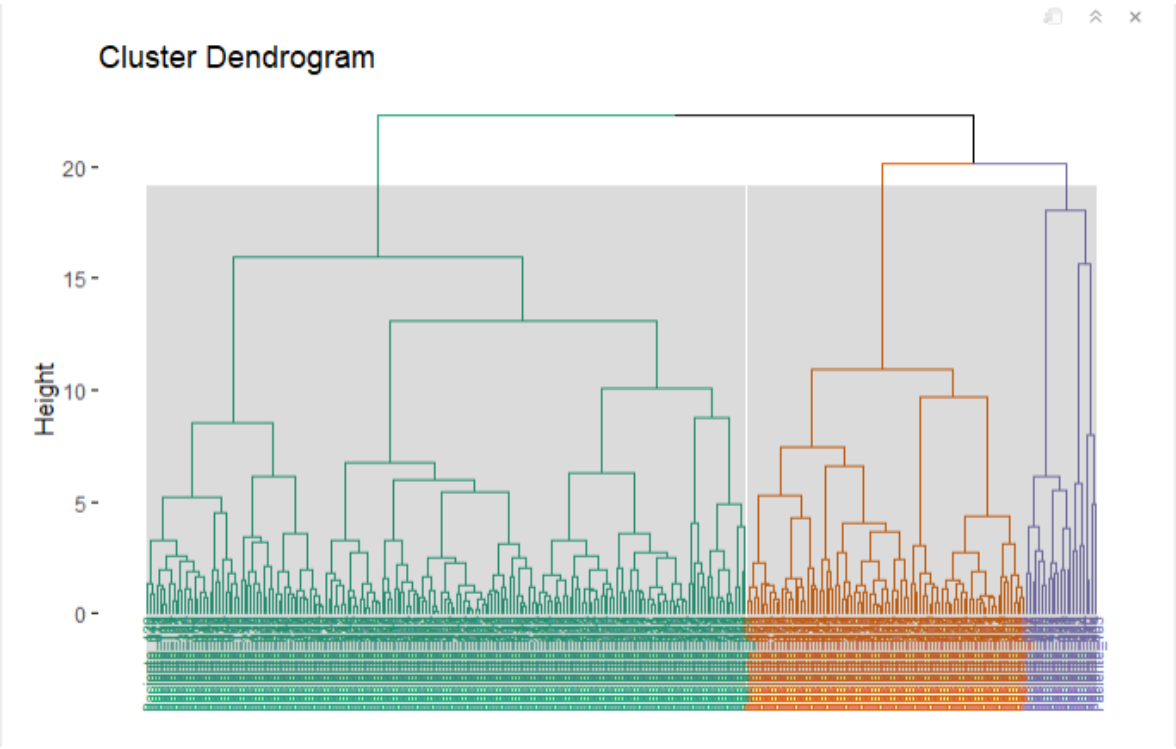
Cada línea en la base representa un paciente, y las ramas que se forman al ascender reflejan cómo se agrupan estos pacientes entre sí.

El árbol crece hacia arriba uniendo los pacientes más parecidos primero, y luego agrupando esos grupos con otros más lejanos.

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_sangre_data,
  k = 3,
  cex = 0.5,
  k_colors = c("#1B9E77", "#D95F02", "#7570B3"),
  color_labels_by_k = TRUE,
  rect = TRUE,
  rect_border = "gray30",
  rect_fill = TRUE
)
```



Interpretación:

Los tres grupos formados muestran una estructura diferenciada y bien separada, lo cual sugiere que existen perfiles clínicos distintos dentro del conjunto de pacientes.

Estos clusters pueden representar, por ejemplo:

Un grupo con pacientes mayores y PSA elevado.

Otro con menor volumen prostático y recurrencia más tardía.

Otro con perfiles intermedios o mixtos.

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.

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).

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)?

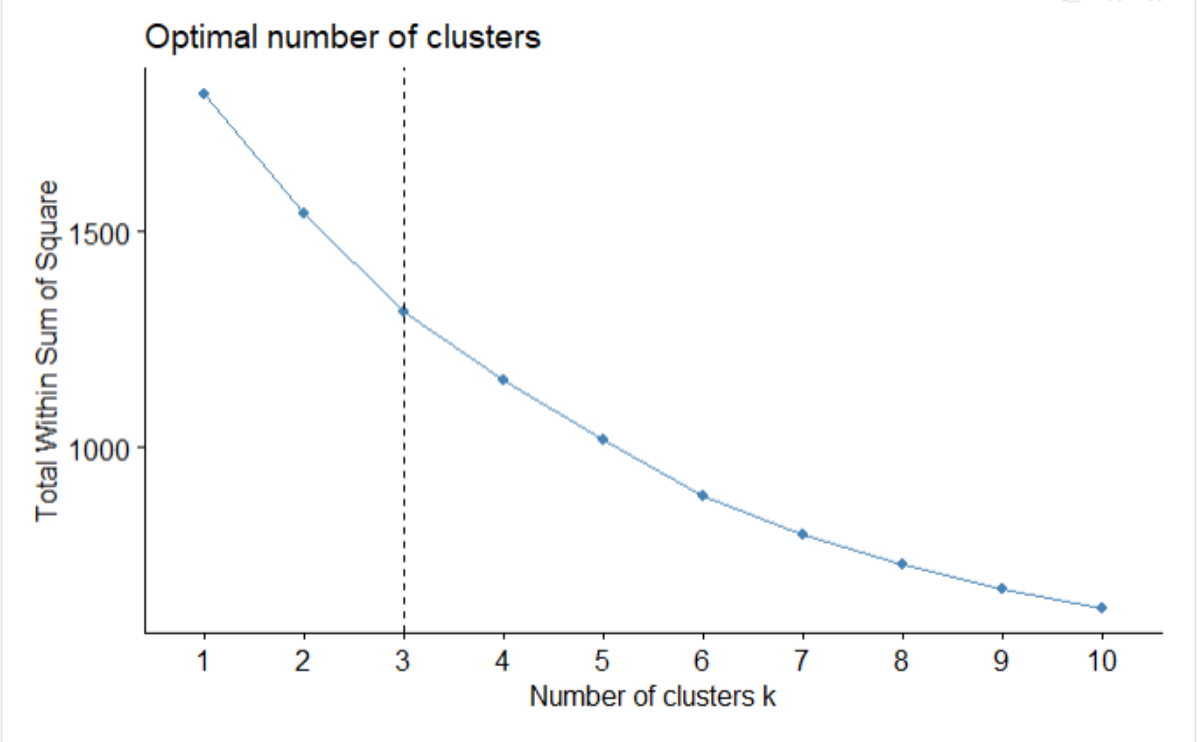
Aquí muestro una solución sencilla y popular: realizar el agrupamiento k-means probando diferentes valores de k (número de clústeres). Luego, se grafica la suma de cuadrados dentro de los clústeres (WSS) en función del número de clústeres. En R, podemos usar la función fviz_nbclust() para estimar el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

Primero escalamos los datos:

```{r}
sangre_data_escalado = scale(sangre_data_1)
```

Ahora graficamos la suma de cuadrados dentro de los gráficos

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



Interpretación:

El "codo" (punto donde la curva deja de descender bruscamente) se encuentra en k = 3, tal como lo indica la línea vertical punteada

Esto sugiere que 3 es el número óptimo de clusters, ya que a partir de ahí agregar más grupos no mejora sustancialmente la compactación interna de los clusters.

El análisis gráfico respalda lo que se observó visualmente en el dendrograma: tres grupos clínicos diferenciados parecen ser una elección adecuada para describir patrones en los pacientes del estudio.

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(sangre_data_escalado, 3, nstart = 25)
```

```{r}
km_res
```

```
K-means clustering with 3 clusters of sizes 182, 28, 94

Cluster means:
         Edad Edad_mediana_GR Volumen_prostata PSA_preoperatorio
Unidades_transfundidas Tiempo_hasta_recurrencia
1  0.02135383    -0.680612391       -0.1041875       -0.26247682
-0.06536776               0.03179719
2  0.27920915    -0.008556663        1.3713996        2.01467679
1.36959493              -0.38575816
3 -0.12451334     1.320330231       -0.2067773       -0.09191669
-0.28140134               0.05334192
```

Interpretación:

- Cluster 1: Compuesto por la mayoría de pacientes, con valores moderados o cercanos al promedio en todas las variables.
- Cluster 2: Perfil muy distintivo. Los pacientes presentan: PSA preoperatorio muy elevado ,Volumen prostático alto ,Más transfusiones ,Tiempo hasta recurrencia más corto (negativo respecto a la media)

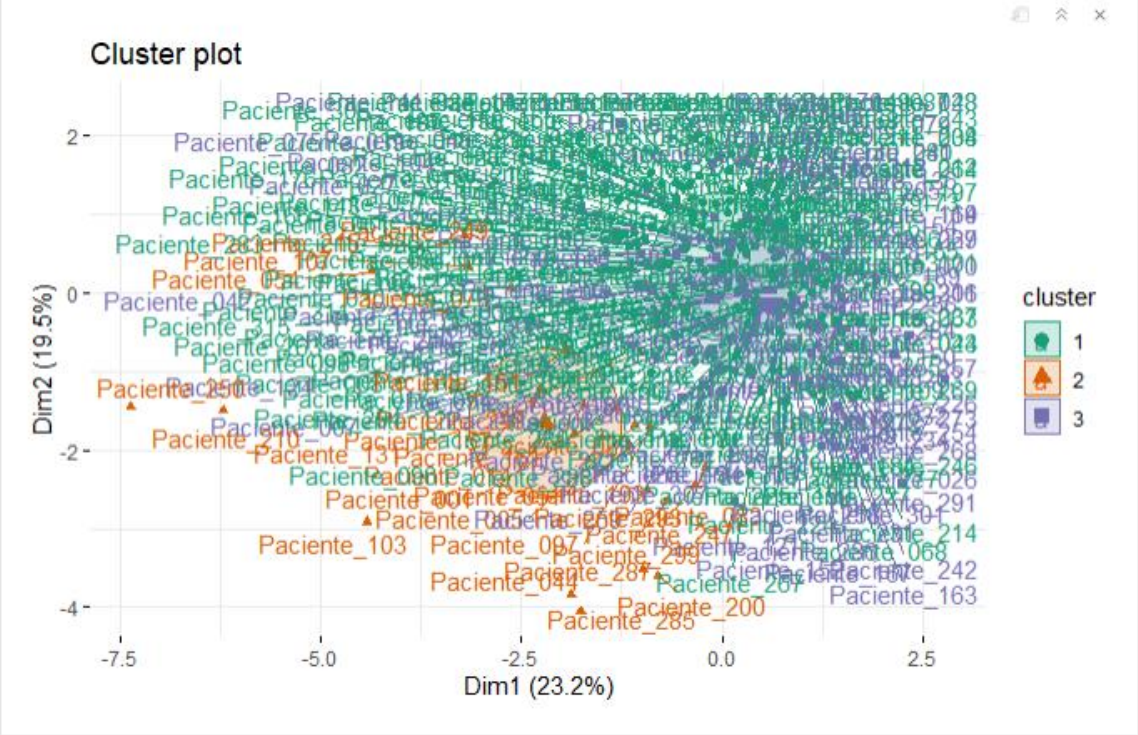
  Este grupo podría representar un subconjunto de alto riesgo o de presentación clínica agresiva.
- Cluster 3: Compuesto por pacientes ligeramente más jóvenes, con características clínicas más cercanas al promedio, pero menor requerimiento de transfusiones. Podría representar un grupo de bajo riesgo relativo.

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 = sangre_data_escalado,
  palette = c("#1B9E77", "#D95F02", "#7570B3"),
  ellipse.type = "euclid",
  repel = TRUE,
  ggtheme = theme_minimal()
)
```



Interpretación:

Cada punto representa un paciente, y el color indica a qué cluster fue asignado:

Verde (Cluster 1): mayoría de los pacientes.

Naranja (Cluster 2): grupo más pequeño y bien separado.

Violeta (Cluster 3): grupo intermedio.

Las dimensiones Dim1 (23.2%) y Dim2 (19.5%) representan combinaciones lineales de tus variables clínicas originales (como PSA, edad, transfusiones, etc.) que explican juntas aproximadamente el 43% de la variabilidad total del conjunto de datos.

Las elipses indican la zona donde se agrupan los pacientes de cada cluster, en base a distancia euclidiana.

Conclusion:

Cluster 1 muestra mayor dispersión, lo que sugiere mayor heterogeneidad clínica o un grupo más general.

Cluster 2 está muy bien delimitado y concentrado, lo que sugiere que sus miembros comparten características clínicas distintivas (como se vio: PSA alto, mayor volumen prostático, más transfusiones).

Cluster 3 también está diferenciado, aunque parcialmente solapado con Cluster 1.