Punto 3

Literal (a)

Genere un conjunto de datos simulados con 20 observaciones en cada una de tres clases (es decir, 60 observaciones en total) y 50 variables.

Sugerencia: *hay una serie de funciones en R que puede utilizar para generar datos. Un ejemplo es la función* **rnorm()***; Otra opción es* **runif()***.Asegúrese de agregar un cambio en la media en las observaciones de cada clase a fin de obtener tres clases distintas.*

Utilizaremos la función `rnorm()` para generar los datos.

```r
set.seed(123)

# Número de observaciones por clase
num_obs <- 20

# Número de variables
num_variables <- 50

# Generar datos para la primera clase con media 0
class1 <- matrix(rnorm(num_obs * num_variables, mean = 0, sd = 1), nrow = num_obs)

# Generar datos para la segunda clase con media 2
class2 <- matrix(rnorm(num_obs * num_variables, mean = 2, sd = 1), nrow = num_obs)

# Generar datos para la tercera clase con media -2
class3 <- matrix(rnorm(num_obs * num_variables, mean = -2, sd = 1), nrow = num_obs)

# Combinar los datos de las tres clases
simulated_data <- rbind(class1, class2, class3)

# Crear un vector de clases (1, 2, 3)
# para identificar a qué clase pertenece cada observación
clases <- rep(1:3, each = num_obs)

# Agregar el vector de clases como una columna a los datos simulados
simulated_data_with_classes <- cbind(clases, simulated_data)

# Convertir la matriz en un data frame para facilitar el manejo
simulated_data_df <- as.data.frame(simulated_data_with_classes)
```

Literal (b)

Realice PCA en las 60 observaciones y grafique las observaciones en términos de las 2 primeras variables principales Z_1 y Z_2. **Use un color diferente** para indicar las observaciones en cada una de las tres clases. Si las tres clases aparecen separados en esta gráfica, **solo** entonces continué con la parte (c). Si no, vuelva al inciso a) y modifique la simulación para que haya una mayor separación entre las tres clases. No continué con la parte (c) hasta que las tres clases muestren al menos algún grado de separación en los dos primeros vectores de scores de componentes principales.

Primero examinamos brevemente los datos.

```
apply(simulated_data_df[,-51], 2, FUN = mean) %>% round(2)
```

```
clases     V2     V3     V4     V5     V6     V7     V8     V9    V10    V11
  2.00   0.13   0.12   0.05  -0.13  -0.02  -0.14  -0.05  -0.15  -0.09   0.00
   V12    V13    V14    V15    V16    V17    V18    V19    V20    V21    V22
  0.02  -0.09  -0.08   0.14   0.04  -0.01   0.00  -0.18   0.16  -0.02  -0.20
   V23    V24    V25    V26    V27    V28    V29    V30    V31    V32    V33
  0.02   0.11   0.00   0.34   0.03   0.16   0.01   0.07  -0.10  -0.07   0.13
   V34    V35    V36    V37    V38    V39    V40    V41    V42    V43    V44
 -0.07  -0.01  -0.27  -0.04   0.03   0.12   0.08   0.01   0.01  -0.05   0.15
   V45    V46    V47    V48    V49    V50
  0.22   0.01   0.08  -0.17   0.22   0.00
```

De la salida anterior notamos que las variables poseen medias muy diferentes.

También podemos examinar las varianzas.

```
apply(simulated_data_df[, -51], 2, FUN = var) %>% round(2)
```

```
clases     V2     V3     V4     V5     V6     V7     V8     V9    V10    V11
  0.68   4.31   3.63   3.40   4.04   4.36   3.51   4.01   3.76   3.81   3.73
   V12    V13    V14    V15    V16    V17    V18    V19    V20    V21    V22
  2.98   3.39   4.63   4.09   3.56   3.29   3.71   3.78   3.32   3.76   4.45
   V23    V24    V25    V26    V27    V28    V29    V30    V31    V32    V33
  3.93   3.88   3.06   3.93   3.42   3.63   4.03   2.83   3.55   4.70   3.88
   V34    V35    V36    V37    V38    V39    V40    V41    V42    V43    V44
  3.28   4.35   4.30   4.54   2.91   4.39   3.78   3.23   3.51   3.85   3.64
   V45    V46    V47    V48    V49    V50
  4.56   3.49   4.48   3.87   3.43   3.62
```

De la salida anterior también se puede observar que las variables tienen varianzas muy diferentes.

Ahora realizaremos el análisis de componentes principales utilizando la función `prcomp()`, la cual es una de varias funciones en R que realizan PCA.

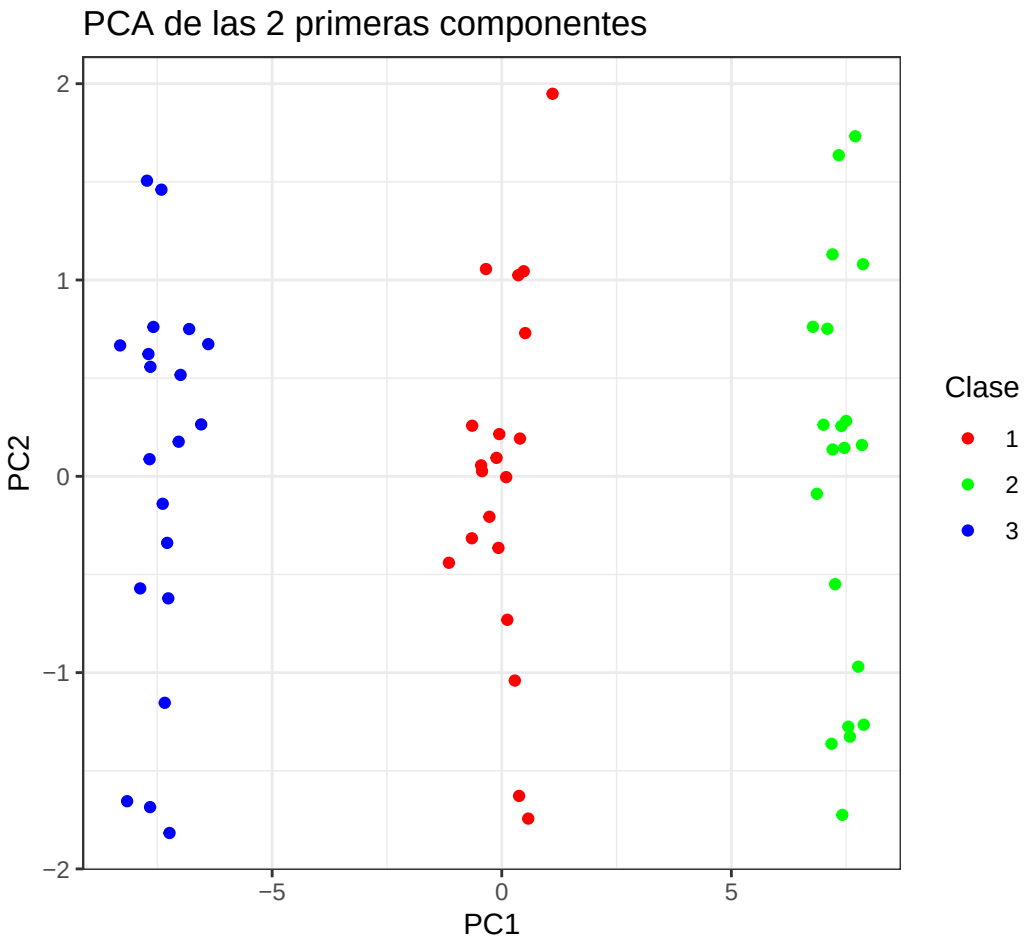


Figure 1: Gráfico de las dos primeras componentes

Observamos en el gráfico anterior Figure 1 tres clases distintas, cada una representada por un color diferente: rojo para la clase 1, verde para la clase 2 y azul para la clase 3. La distribución de los puntos sugiere que el PCA ha logrado una separación relativa entre las clases, especialmente entre la clase 3 y las otras dos clases, en las direcciones de las dos primeras componentes principales.

Literal (c)

Desarrolle agrupación de K-medias de las observaciones con K = 3. ¿Que tan bien funcionan los clústeres que obtuvo con el algoritmo de K-medias comparado con las verdaderas etiquetas de clase?

Sugerencia: *puede usar la función* `table()` *en R para comparar las verdaderas etiquetas de clase con las etiquetas de clase obtenidas por agrupamiento. Tener cuidado como se interpretan los resultados: el agrupamiento de K-medias numera los grupos arbitrariamente, por lo que no puede simplemente comprobar si las verdaderas etiquetas de clase y las etiquetas de agrupación son las mismas.*

```r
# Realizar agrupación de K-means, con k=3
set.seed(123)
kmeans_result <- kmeans(simulated_data_df[, -1], centers = 3, nstart = 20)

comparison_table <- table("Clase Verdadera" = simulated_data_df$clases,
                          "Cluster Asignado" = kmeans_result$cluster)
comparison_table
```

```
               Cluster Asignado
Clase Verdadera  1  2  3
              1  0 20  0
              2 20  0  0
              3  0  0 20
```

La tabla indica que cada una de las clases verdaderas ha sido asignada a un cluster distinto, sin mezclas entre ellas. Esto significa que los 20 elementos de la clase verdadera 1 han sido asignados al cluster 2, los 20 elementos de la clase verdadera 2 al cluster 1, y los 20 elementos de la clase verdadera 3 al cluster 3.

El hecho de que no haya mezcla entre las clases en los clusters asignados por K-medias sugiere que el algoritmo ha realizado una perfecta clasificación de los datos en base a sus características, asignando cada observación a un cluster correspondiente a su clase verdadera. La interpretación debe tener en cuenta que los números de los clusters asignados por K-medias son arbitrarios y no tienen por qué coincidir con las etiquetas de las clases verdaderas, por lo que una coincidencia directa no es necesaria para que la clasificación sea correcta.

Literal (d)

Realice agrupamiento de K-medias con K = 2. Describa sus resultados.

```r
# Realizar agrupación de K-means, con k=2
set.seed(123)
kmeans_result <- kmeans(simulated_data_df[, -1], centers = 2, nstart = 20)

comparison_table <- table("Clase Verdadera" = simulated_data_df$clases,
                          "Cluster Asignado" = kmeans_result$cluster)
comparison_table
```

```
               Cluster Asignado
Clase Verdadera  1  2
              1  0 20
              2  0 20
              3 20  0
```

Este resultado indica que el algoritmo de K-medias ha agrupado juntas las Clases Verdaderas 1 y 2 en el mismo cluster (Cluster 2), mientras que la Clase Verdadera 3 ha sido separada en un cluster diferente (Cluster 1). Dado que K-medias con $K = 2$ solo permite dos grupos, el algoritmo ha combinado las dos clases más similares entre sí en términos de la distancia de sus características.

Literal (e)

Ahora realice agrupamiento de K-medias con K = 4 y describa su resultados.

```r
# Realizar agrupación de K-means, con k=4
set.seed(123)
kmeans_result <- kmeans(simulated_data_df[, -1], centers = 4, nstart = 20)

comparison_table <- table("Clase Verdadera" = simulated_data_df$clases,
                          "Cluster Asignado" = kmeans_result$cluster)
comparison_table
```

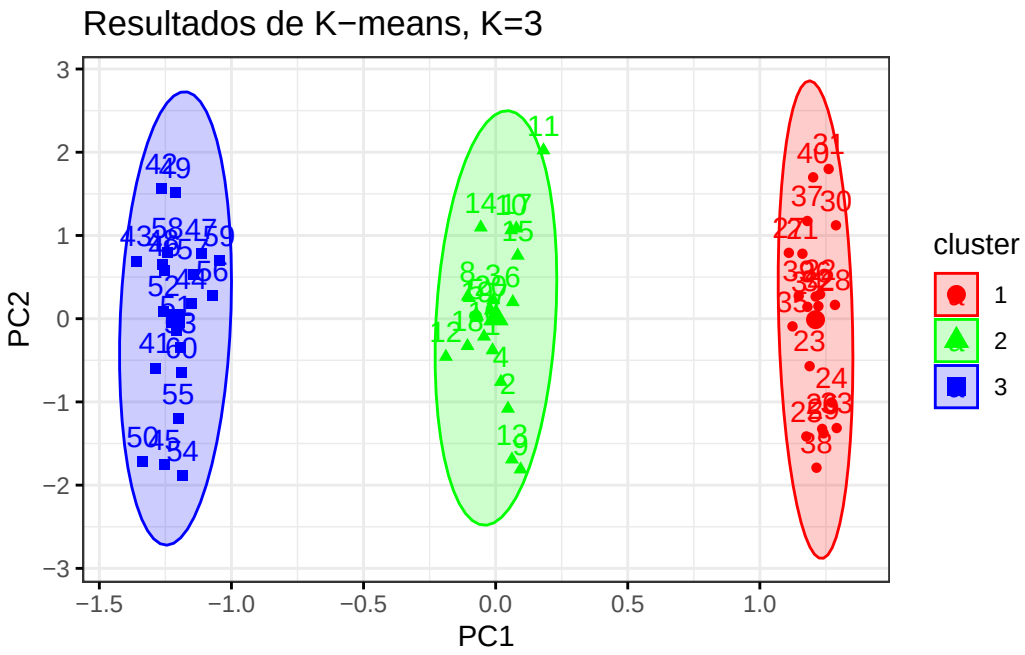
```
               Cluster Asignado
Clase Verdadera  1  2  3  4
              1  0  0 20  0
              2  8  0  0 12
              3  0 20  0  0
```

Este resultado sugiere que mientras las Clases Verdaderas 1 y 3 han sido identificadas y segregadas claramente en clusters únicos (Clusters 3 y 2 respectivamente), la Clase Verdadera 2 ha sido repartida entre dos clusters. Esto puede indicar que, desde la perspectiva del algoritmo y basado en la medida de distancia utilizada (usualmente la distancia euclidiana), las observaciones de la Clase Verdadera 2 no son tan cohesivas como las de las Clases Verdaderas 1 y 3, y quizás presentan una mayor variabilidad interna o tienen características que las solapan con las variabilidades de las otras clases.

El hecho de que haya una división en la Clase Verdadera 2 podría reflejar una estructura subyacente no capturada por las etiquetas de clase originales o podría ser una señal de que el valor de K escogido no es el más adecuado para este conjunto de datos. En la práctica, elegir el valor correcto de K a menudo implica un compromiso entre sobresimplificar la estructura de los datos (un valor de K muy bajo) y sobreajustar el ruido (un valor de K muy alto). Herramientas como el método del codo o la silueta pueden ayudar a determinar el valor más apropiado de K.

Literal (f)

Ahora realice agrupamiento de K-medias con K = 3 en los dos primeros vectores de scores de componentes principales, en lugar de los datos en las variables originales. Es decir, realice la agrupación de K-medias en la matriz de tamaño 60×2, cuya primera columna es la coordenada z_{i1} en la primera componente principal Z_1 y la segunda columna es la coordenada z_{i2} en la segunda componente principal Z_2. **Comente los resultados**.



El gráfico ilustra cómo se distribuyen las observaciones en el espacio de las dos primeras componentes principales (PC1 y PC2), que son combinaciones lineales de las variables originales, seleccionadas de tal manera que maximizan la varianza de los datos. Al aplicar K-means a las componentes principales en lugar de las variables originales, se está agrupando los datos basándose en las direcciones de mayor varianza, lo cual puede llevar a una mejor separación de los clusters si las componentes principales capturan bien la estructura de los datos.

Es de resaltar que la separación clara entre los clusters sugiere que el agrupamiento ha sido efectivo en este espacio de componentes principales. Esto es especialmente útil si las variables originales tienen una correlación alta o si el conjunto de datos es de alta dimensión, ya que el PCA reduce la redundancia antes de aplicar K-medias, lo que puede mejorar la calidad de los clusters encontrados. Además, trabajar con componentes principales en lugar de con las variables originales puede hacer que la interpretación de los resultados sea más sencilla y más resistente a ciertos tipos de ruido y variabilidad irrelevante en los datos.

Literal (g)

Con la función `scale()`, realice agrupamiento de K-medias con K = 3 en los datos después de escalar cada variable para tener una desviación estándar de uno. ¿Como se comparan estos resultados con los obtenidos? en (b)? Explique.

```
# Escalar los datos
scaled_data <- scale(simulated_data_df[, -1])

# Realizar agrupación de K-medias en los datos escalados
set.seed(123)
kmeans_scaled_result <- kmeans(scaled_data, centers = 3)

comparison_table_scaled <- table("Clase Verdadera" = simulated_data_df$clases,
```

```
                                  "Cluster Asignado" = kmeans_scaled_result$cluster)

comparison_table_scaled
```

```
               Cluster Asignado
Clase Verdadera  1  2  3
              1  0 20  0
              2 20  0  0
              3  0  0 20
```

Comparando estos resultados con los obtenidos en (b), podemos observar que en ambos casos el algoritmo de K-medias logró un agrupamiento perfecto. Sin embargo, la diferencia clave entre los dos métodos es el enfoque de la transformación de los datos:

- el caso de la PCA, se realizó una reducción de dimensionalidad, donde se transformaron las variables originales en componentes principales que capturan la mayor variación posible en los datos.
- En el caso de la estandarización con `scale()`, se ajustó la escala de las variables originales para que todas contribuyeran equitativamente al análisis de K-medias, eliminando el sesgo que podrían introducir las variables con mayor varianza.

La consistencia de los resultados entre los dos métodos sugiere que las clases verdaderas tienen diferencias significativas que se pueden detectar tanto en las direcciones de máxima varianza (PCA) como en el espacio de las variables originales después de la estandarización. Esto también indica que las clases son inherentemente bien separables y que los métodos de preprocesamiento aplicados (PCA y estandarización) han sido efectivos para revelar esta separabilidad en el espacio de características transformado.