Análisis de Datos I Tarea 3

Maria Carolina Navarro Monge C05513 Tábata Picado Carmona C05961
Jose Pablo Trejos Conejo C07862

Se cargan las librerías necesarias.

```
library(readr)
library(FactoMineR)
library(ggplot2)
library(dplyr)
library(gridExtra)
library(factoextra)
```

Ejercicio 1

Complete todas las demostraciones que quedaron como tarea en la presentación de la clase.

Para el desarrollo de los siguientes ejercicios se emplea la siguiente matriz:

```
X <- matrix(c(8,1,0,4,6,5,6,8,7,10,4,7,8,2,5,0,3,6), nrow = 6, ncol = 3, byrow
            = TRUE)
```

Además, para el ejercicio 8 se utilizan las siguientes bases de datos:

```
#----- Estudiantes--------
estudiantes_datos <- read.table('EjemploEstudiantes.csv', header=TRUE, sep=';',
                                dec=',',row.names=1)
estudiantes_datos<- as.matrix(estudiantes_datos)

#----- beans--------------
beans_datos <- read.csv("beansV2.csv")

#Convertir Class a código disyuntivo
categorias_class  <- unique(beans_datos$Class)
lista_disyuntivo  <- list()
beans_datos_disyuntivo <- beans_datos[,-17]

for (i in 1:length(categorias_class)) {
  lista_disyuntivo[[i]] <- as.numeric(beans_datos$Class == categorias_class[i])
  names(lista_disyuntivo)[i] <- paste("Class_", categorias_class[i], sep = "")
  beans_datos_disyuntivo <- cbind(beans_datos_disyuntivo, lista_disyuntivo[i])
}
beans_datos <- as.matrix(beans_datos_disyuntivo)
```

Ejercicio 2

Implemente en lenguaje R funciones que permitan ejecutar el algoritmo del Análisis en Componentes Principales (ACP) visto en clase para variables numéricas. Compare los resultados obtenidos con respecto a FactoMineR

Primeramente, se construyen las siguientes funciones necesarias para plantear una función que realice un ACP de una base de datos.

```r
#Centramos y reducimos la matriz con la siguiente función
centrar_y_reducir <-  function(matriz){
  # Se obtiene la cantidad de filas y columnas que posee la matriz.
  columnas <- ncol(matriz)
  filas <- nrow(matriz)

  # Se crea una nueva matriz con los datos centrados y reducidos.
  for(i in 1:columnas){
    matriz[,i] <- ((matriz[,i] - mean(matriz[,i])) /
                     (sd(matriz[,i])*sqrt((filas - 1)/filas)))
  }

  return(matriz)
}
```

```r
#Calcular la matriz de correlaciones
matriz_correlaciones_ind <- function(matriz) {
  n <- nrow(matriz)
  resultado <- (1/n)*t(matriz)%*%matriz
  return(resultado)
}
```

```r
#Calcular la matriz de calidades de los individuos (cosenos cuadrados)
matriz_cos2_ind <- function(C, matriz) {
  n <-nrow(matriz)
  m <-ncol(matriz)
  resultado <- matrix(0, n,m)

  for(i in 1: n){
    suma <- 0
    for(j in 1: m){
      suma <- suma + matriz[i,j]^2
    }
    for(r in 1: m){
      resultado[i,r] <- (C[i,r]^2)/suma
    }
  }
  return(resultado)
}
```

```r
#Calcular la contribución de cada individuo i a la varianza total del eje r
contribucion_ind <- function(C, valores_propios) {
  n <-nrow(C)
  m <-ncol(C)
  resultado <- matrix(0, n, m)
```

```
  for(i in 1: n){
    for(r in 1: m){
      resultado[i,r] <- ((C[i,r]^2)/(n*valores_propios[r]))*100
    }
  }
  return(resultado)
}
```

```
# Calcular la matriz de coordenadas de las variables
coordenadas_var <- function(V, valores_propios){
  m <- ncol(V)
  resultado <- matrix(0, m, m)

  for(r in 1:m){
      resultado[,r] <- sqrt(valores_propios[r])*V[,r]
    }

  return(resultado)
}
```

```
# Obtener contribuciones de las variables al cálculo de la varianza total
# del eje r.
contribuciones_var <- function(cos2){
  m <- ncol(cos2)
  contribuciones <- matrix(nrow = m, ncol = m)

  for(i in 1:m){
    for(j in 1:m){
      contribuciones[j,i] <- (cos2[j,i]/sum(cos2[,i]))*100
    }
  }

  return(contribuciones)
}
```

```
#Calcular vector de inercias de los ejes
calcular_inercias <- function(valores_propios){
  m <- length(valores_propios)
  resultado <- c()

  for(j in 1: m){
    resultado[j] <-100*(valores_propios[j]/m)
  }
  return(resultado)
}
```

Usamos la funciones anteriores para construir una sola función que retorne el ACP de una base de datos.

```
ACP_funcion <- function(X){
  #1) centrar y reducir X
  X <- centrar_y_reducir(X)
```

```
  #2) matriz de correlaciones
  X_R <- matriz_correlaciones_ind(X)

  #3y 4) valores y vectores propios de la matriz de correlaciones
  X_R_e <- eigen(X_R)
  X_R_valores_propios <- X_R_e$values #ya vienen ordenados de mayor a menor
  X_R_vectores_propios <- X_R_e$vectors
  X_R_vectores_propios

  #5) Construir matriz V que tiene como columnas los vectores propios de la
  #matriz de correlaciones
  V <- X_R_vectores_propios

  #6) Calcular la matriz de componentes principales
  C <- X%*%V

  #7) Calcular la matriz de calidades de los individuos
  X_Q <- matriz_cos2_ind(C, X)

  # La matriz de contribuciones para los individuos de X es:
  X_contrib <- contribucion_ind(C, X_R_valores_propios)

  #8) La matriz de coordenadas de las variables
  X_T <- coordenadas_var(V, X_R_valores_propios)

  #9) Matriz de calidades de las variables
  S <- X_T^2

  # Matriz de contribuciones de las variables
  var_contrib <- contribuciones_var(S)

  #10) Vector de inercias
  X_I <- calcular_inercias(X_R_valores_propios)

  resultado <- list(valores_propios = X_R_valores_propios,
                    ind = list(coord = C, cos2 =   X_Q, contrib = X_contrib),
                    var = list(coord = X_T, cos2 = S, contrib = var_contrib),
                    inercias = X_I)
}
```

Comparamos los resultados obtenidos con los de FactoMiner para la matriz X.

Valores propios

```
ACP_X <- ACP_funcion(X)
ACP_X$valores_propios
```

```
## [1] 1.7278743 0.9191845 0.3529412
```

Comparamos con lo obtenido con FactoMiner

```r
X_ACP <- PCA(X, ncp = 4, graph = FALSE)
X_ACP$eig
```

```
##        eigenvalue percentage of variance cumulative percentage of variance
## comp 1  1.7278743              57.59581                          57.59581
## comp 2  0.9191845              30.63948                          88.23529
## comp 3  0.3529412              11.76471                         100.00000
```

Se puede observar que los valores propios obtenidos con FactoMiner son iguales a los del algoritmo.

Coordenadas de los individuos

```r
ACP_X$ind$coord
```

```
##            [,1]       [,2]       [,3]
## [1,]  2.4480754 -0.1702518 -0.5940885
## [2,] -0.7572077 -0.3931637 -0.5940885
## [3,] -1.6908677  0.5634155 -0.5940885
## [4,] -0.1764523  1.3497428  0.5940885
## [5,]  0.7572077  0.3931637  0.5940885
## [6,] -0.5807554 -1.7429065  0.5940885
```

Con FactoMiner se obtiene lo siguiente:

```r
X_ACP$ind$coord
```

```
##        Dim.1      Dim.2      Dim.3
## 1 -2.4480754 -0.1702518  0.5940885
## 2  0.7572077 -0.3931637  0.5940885
## 3  1.6908677  0.5634155  0.5940885
## 4  0.1764523  1.3497428 -0.5940885
## 5 -0.7572077  0.3931637 -0.5940885
## 6  0.5807554 -1.7429065 -0.5940885
```

Podemos observar que lo único que varía es el signo de algunas entradas, pero esto solo indica que se refleja con respecto a algunos de los ejes al graficar.

Matriz de cosenos al cuadrado de los individuos

```r
ACP_X$ind$cos2
```

```
##            [,1]        [,2]       [,3]
## [1,] 0.94008991 0.004546772 0.05536332
## [2,] 0.53045874 0.143010648 0.32653061
## [3,] 0.81005952 0.089940481 0.10000000
## [4,] 0.01411472 0.825885276 0.16000000
## [5,] 0.53045874 0.143010648 0.32653061
## [6,] 0.09047267 0.814852770 0.09467456
```

Vemos los resultados de FactoMiner

```
X_ACP$ind$cos2
```

```
##        Dim.1       Dim.2      Dim.3
## 1 0.94008991 0.004546772 0.05536332
## 2 0.53045874 0.143010648 0.32653061
## 3 0.81005952 0.089940481 0.10000000
## 4 0.01411472 0.825885276 0.16000000
## 5 0.53045874 0.143010648 0.32653061
## 6 0.09047267 0.814852770 0.09467456
```

Los cuales son iguales a los obtenidos con el algoritmo.

Constribución de los individuos

```
ACP_X$ind$contrib
```

```
##            [,1]       [,2]     [,3]
## [1,] 57.8077648  0.5255686 16.66667
## [2,]  5.5305285  2.8028049 16.66667
## [3,] 27.5775612  5.7557722 16.66667
## [4,]  0.3003249 33.0330084 16.66667
## [5,]  5.5305285  2.8028049 16.66667
## [6,]  3.2532922 55.0800411 16.66667
```

Con FactoMiner se tiene:

```
X_ACP$ind$contrib
```

```
##        Dim.1      Dim.2    Dim.3
## 1 57.8077648  0.5255686 16.66667
## 2  5.5305285  2.8028049 16.66667
## 3 27.5775612  5.7557722 16.66667
## 4  0.3003249 33.0330084 16.66667
## 5  5.5305285  2.8028049 16.66667
## 6  3.2532922 55.0800411 16.66667
```

Por lo tanto, se tiene el mismo resultado que FactoMiner.

Coordenadas de las variables La matriz de coordenadas de las variables para los datos que tenemos es:

```
ACP_X$var$coord
```

```
##            [,1]      [,2]          [,3]
## [1,]  0.4155396 0.9095751 -6.164154e-17
## [2,] -0.8818166 0.2143101 -4.200840e-01
## [3,] -0.8818166 0.2143101  4.200840e-01
```

Y la dada con FactoMiner es:

```
X_ACP$var$coord
```

```
##         Dim.1     Dim.2     Dim.3
## V1 -0.4155396 0.9095751  0.000000
## V2  0.8818166 0.2143101  0.420084
## V3  0.8818166 0.2143101 -0.420084
```

Matriz de cosenos al cuadrado de las variables

```
ACP_X$var$cos2
```

```
##           [,1]       [,2]         [,3]
## [1,] 0.1726732 0.82732684 3.799680e-33
## [2,] 0.7776006 0.04592883 1.764706e-01
## [3,] 0.7776006 0.04592883 1.764706e-01
```

Con FactoMiner da:

```
X_ACP$var$cos2
```

```
##        Dim.1      Dim.2     Dim.3
## V1 0.1726732 0.82732684 0.0000000
## V2 0.7776006 0.04592883 0.1764706
## V3 0.7776006 0.04592883 0.1764706
```

Contribución de las variables

```
ACP_X$var$contrib
```

```
##           [,1]      [,2]         [,3]
## [1,]  9.993387 90.006613 1.076576e-30
## [2,] 45.003307  4.996693 5.000000e+01
## [3,] 45.003307  4.996693 5.000000e+01
```

Con FactoMiner se obtiene

```
X_ACP$var$contrib
```

```
##        Dim.1     Dim.2 Dim.3
## V1  9.993387 90.006613     0
## V2 45.003307  4.996693    50
## V3 45.003307  4.996693    50
```

Por temas computacionales, el primer valor de la tercera dimensión de las matrices de coordenadas, calidades y contribuciones de las variables, obtenidas con el algoritmo, es muy cercano a cero. Por lo cual, con FactoMiner aparece como cero.

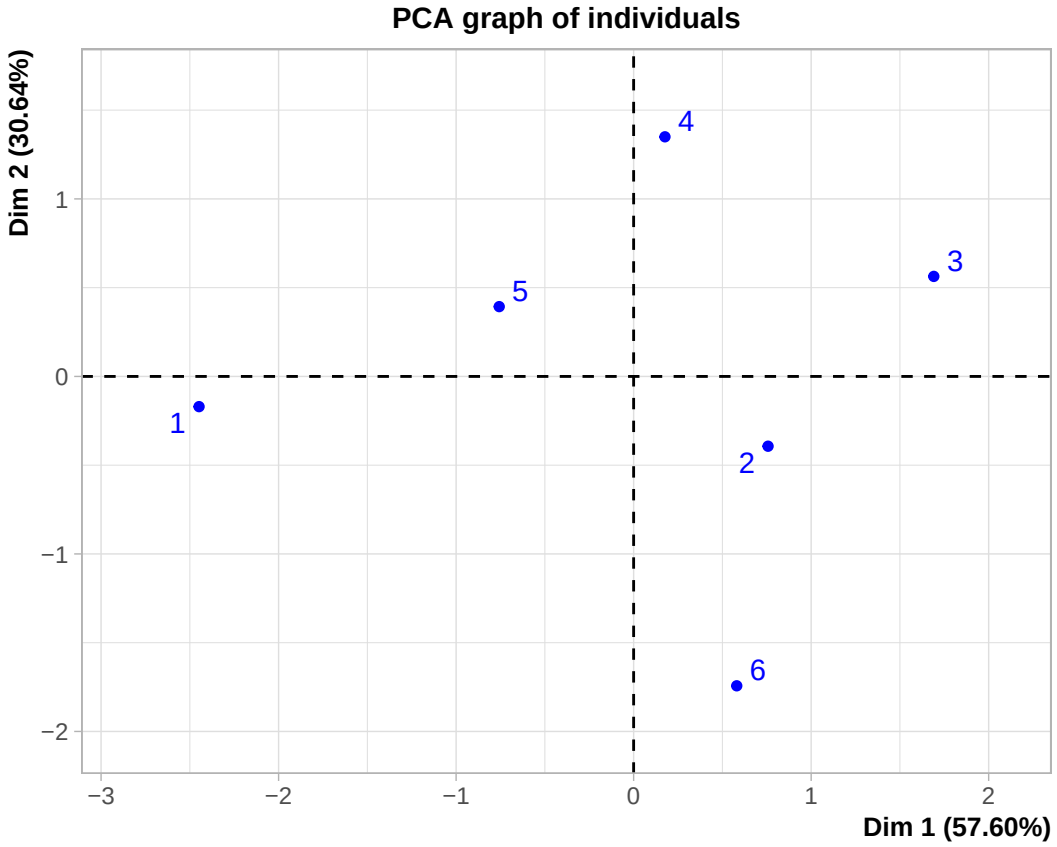
Inercias

```
ACP_X$inercias[1:2]
```

```
## [1] 57.59581 30.63948
```

Con FactoMiner se tiene:

```
plot(X_ACP, axes=c(1, 2), choix="ind", col.ind="blue",new.plot=TRUE)
```



Podemos observar que las inercias correpondiente al eje "x" y "y" obtenidas con FactoMiner son iguales a las dadas por el algoritmo.

Ejercicio 3

Implemente en lenguaje R funciones que grafiquen el plano principal (en 2 dimensiones),el círculo de correlaciones (en 2 dimensiones) que incluya la inercia, el gráfico dual (en 2D) para ver ambos gráficos juntos. Compare los resultados obtenidos con respecto a FactoMineR

A continuación se presentan las funciones para calcular dichos gráficos

```
#---------------------------Plano principal individuos--------------------------

plano_principal <- function(matriz){
  #Obtenemos la matriz con las coordenadas de los individuos

  ACP_matriz <- ACP_funcion(matriz)
```

```r
  C <- ACP_matriz$ind$coord

  #Convertimos la matriz en un dataframe
  C_data <- as.data.frame(C)
  col_names <- paste("Dim", 1:ncol(C_data))
  colnames(C_data) <- col_names

  if(is.null(rownames(C_data))) {
    C_data$individuo <- seq_len(nrow(C_data))
  } else {
    C_data$individuo <- rownames(C_data)
  }

  #Sacamos la inercia
  X_I <- ACP_matriz$inercias
  X_I_redondeados <- round(X_I, 2)
  inercia <- X_I_redondeados[1] + X_I_redondeados[2]


  #Graficamos

  if(nrow(C_data) <= 10){
    individuos <- ggplot(C_data, aes(x = `Dim 1`, y = `Dim 2`)) +
      geom_point(color = "lightblue") +
      geom_text(aes(label = individuo), vjust = 0, hjust = 0,
                color = "lightblue") +
      labs(x = paste("Dim 1 (", X_I_redondeados[1], "%)"),
         y = paste("Dim 2 (", X_I_redondeados[2], "%)"),
         subtitle = paste("Inercia = ", inercia, "%"),
         title = "Plano principal (Individuos)")+
      theme_minimal()
  }else{
    individuos <- ggplot(C_data, aes(x = `Dim 1`, y = `Dim 2`)) +
      geom_point(color = "lightblue") +
      labs(x = paste("Dim 1 (", X_I_redondeados[1], "%)"),
         y = paste("Dim 2 (", X_I_redondeados[2], "%)"),
         subtitle = paste("Inercia = ", inercia, "%"),
         title = "Plano principal (Individuos)") +
      theme_minimal()
  }

  return(individuos)
}
```

```r
#----------------------------Círculo de correlaciones----------------------------
circulo_correlaciones <- function(matriz){
  #Obtenemos la matriz con las coordenadas de las variables

  ACP_matriz <- ACP_funcion(matriz)

  X_T <- ACP_matriz$var$coord

  #Convertimos la matriz en un dataframe y ajustamos para el gráfico
```

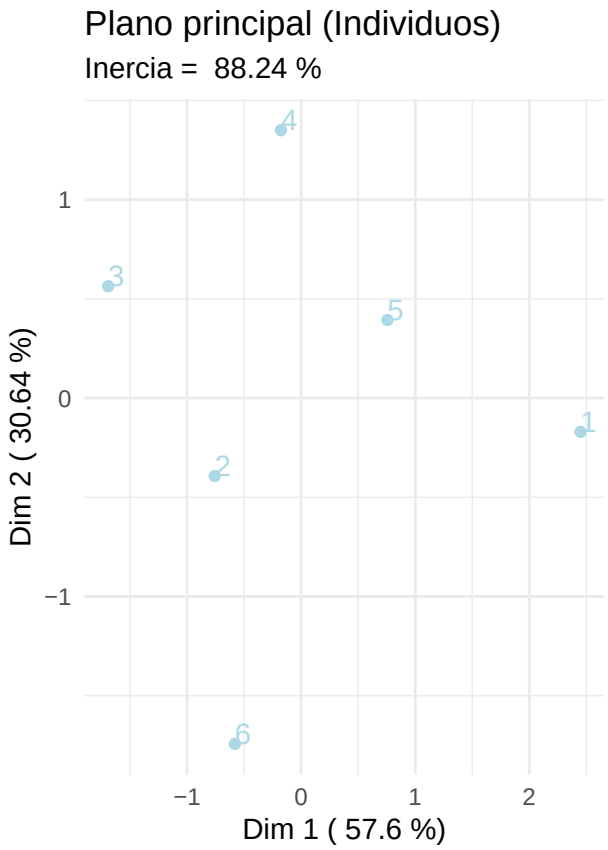
```
  X_T_data <- as.data.frame(X_T)

  col_names <- paste("Dim", 1:ncol(X_T_data))
  colnames(X_T_data) <- col_names

  if(is.null(colnames(matriz))) {
    X_T_data$variable <- seq_len(ncol(matriz))
  } else {
    X_T_data$variable <- colnames(matriz)
  }
  X_T_data$`x origen` <- 0
  X_T_data$`y origen` <- 0

  #Sacamos la inercia
  X_I <- ACP_matriz$inercias
  X_I_redondeados <- round(X_I, 2)
  inercia <- X_I_redondeados[1] + X_I_redondeados[2]

  #Graficamos
  variables0 <- ggplot(X_T_data, aes(x = `x origen`, y = `y origen`)) +
    geom_segment(aes(xend = `Dim 1`, yend = `Dim 2`),
                 arrow = arrow(length = unit(0.2, "inches")), color = "orange")+
    geom_text(aes(x = `Dim 1`, y = `Dim 2`, label = variable), vjust = -0.5,
              nudge_y = 0, nudge_x = 0, color ="orange") +
    labs(x = paste("Dim 1 (", X_I_redondeados[1], "%)"),
         y = paste("Dim 2 (", X_I_redondeados[2], "%)"),
         subtitle = paste("Inercia = ", inercia, "%")) +
    theme_minimal()

  variables <- variables0 +
    geom_path(data = data.frame(x = cos(seq(0, 2 * pi, length.out = 100)),
                                y = sin(seq(0, 2 * pi, length.out = 100))),
              aes(x, y), color = "black", linewidth = 1, linetype = "dashed") +
    labs(title = "Círculo de correlaciones")

  return(list(variables0 = variables0, variables = variables))
}
```

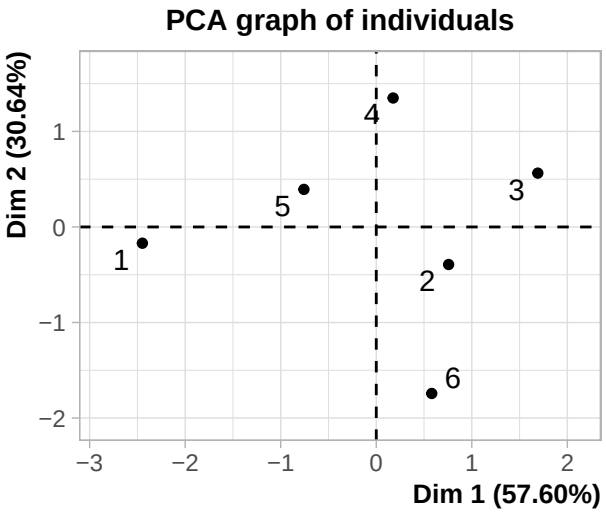
```
#-------------------------Grafico Dual----------------------------------------
grafico_dual <- function(graf_ind, graf_var, matriz) {
  # Obtener los graficos y las etiquetas
  grafico_circulo <- graf_var$variables0
  data_graf_ind <- ggplot_build(graf_ind)$data[[1]]
  data_graf_var <-ggplot_build(grafico_circulo)$data[[1]]

  if(is.null(rownames(matriz))) {
    data_graf_ind$label <- seq_len(nrow(matriz))
  } else {
    data_graf_ind$label <- rownames(matriz)
  }

  if (is.null(colnames(matriz))){
    data_graf_var$label <- seq_len(ncol(matriz))
```

```
  }else {
    data_graf_var$label <- colnames(matriz)
  }

  # Superponer los gráficos
  if(nrow(data_graf_ind) <= 10){
    grafico_final <- grafico_circulo +
      geom_point(data = data_graf_ind, aes(x, y), color = "lightblue") +
      geom_text(data = data_graf_ind, aes(x, y, label = label),
                vjust = 0, hjust = 0, color = "lightblue")
  }else{
    #grafico_final <- grafico_circulo +
      #geom_point(data = data_graf_ind, aes(x, y), color = "lightblue", size=1)
    grafico_final <- graf_ind +
      geom_segment(data = data_graf_var, aes(x = x, y = y, xend =  xend,
                                             yend = yend),
                   arrow = arrow(length = unit(0.2, "inches")),
                   color = "orange") +
      geom_text(data = data_graf_var, aes(x = xend, y = yend, label =label),
                vjust = 0, hjust = 0, color = "orange" )
  }

  return(grafico_final)
}
```

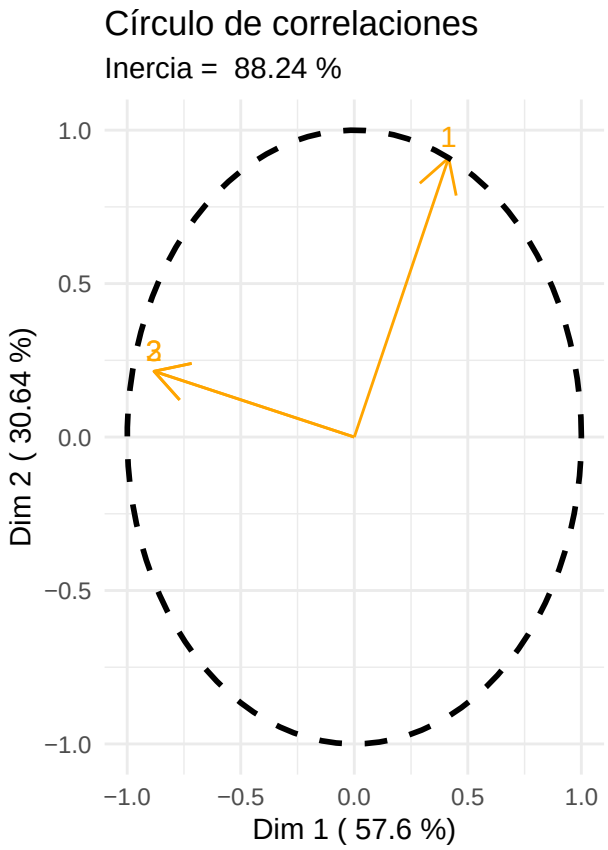
Ahora, comparamos los gráficos obtenidos con el algoritmo para la matriz X y los dados por FactoMiner.

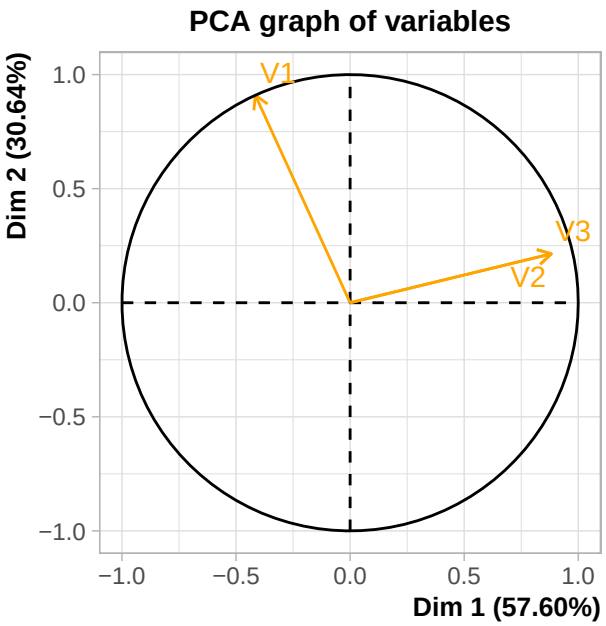
```
individuos_FM <- plot(X_ACP)

grid.arrange(plano_principal(X), individuos_FM, ncol = 2)
```



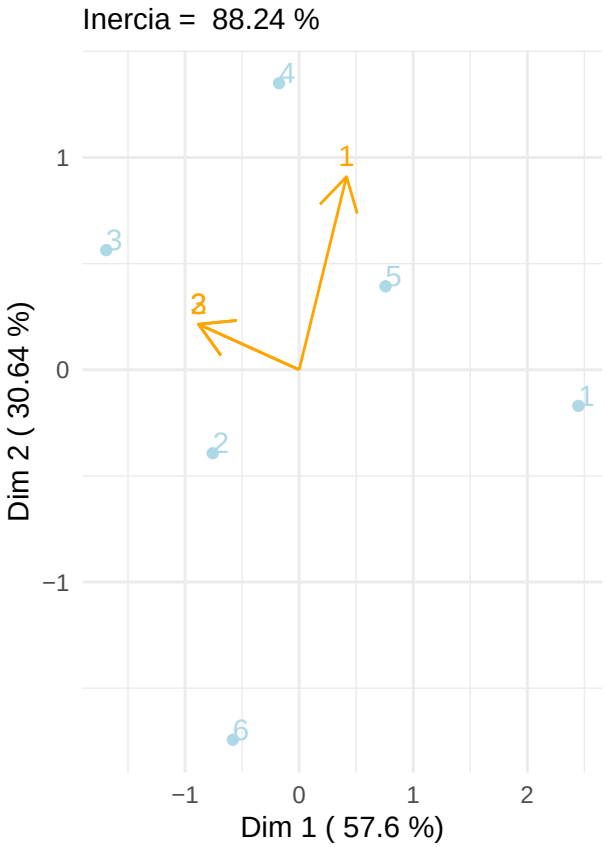


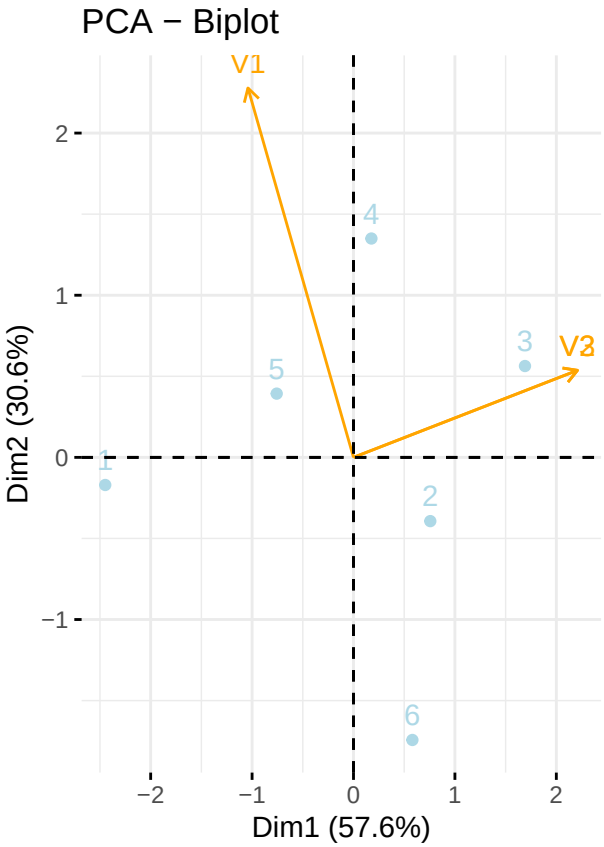

```
variables_FM <- plot(X_ACP, axes=c(1, 2), choix="var", col.var="orange",
                     new.plot=TRUE)

grid.arrange(circulo_correlaciones(X)[["variables"]], variables_FM, ncol = 2)
```






```
dual_FM <- fviz_pca_biplot(X_ACP,col.var = "orange",col.ind = "lightblue")
grid.arrange(grafico_dual(plano_principal(X), circulo_correlaciones(X), X),
             dual_FM, ncol = 2)
```





Como se puede observar, todos los gráficos generados por FactoMiner están reflejados con respecto al eje "y" en comparación con los generados por el algortimo .

Ejercicio 4

Programe una función en R que reciba una fila (individuo) de una matriz y calcule la proyección en suplementario de este individuo en el plano principal 2D programado en el punto anterior. Compare los resultados obtenidos con respecto a FactoMineR

Primeramente, es necesario mencionar que el individuo suplementario se considera como una nueva fila que se le añade a la base de datos inicial. Por lo tanto, en caso de que la fila no tenga una etiqueta, se le coloca un identificador que corresponde al número de fila, en este caso, el último.

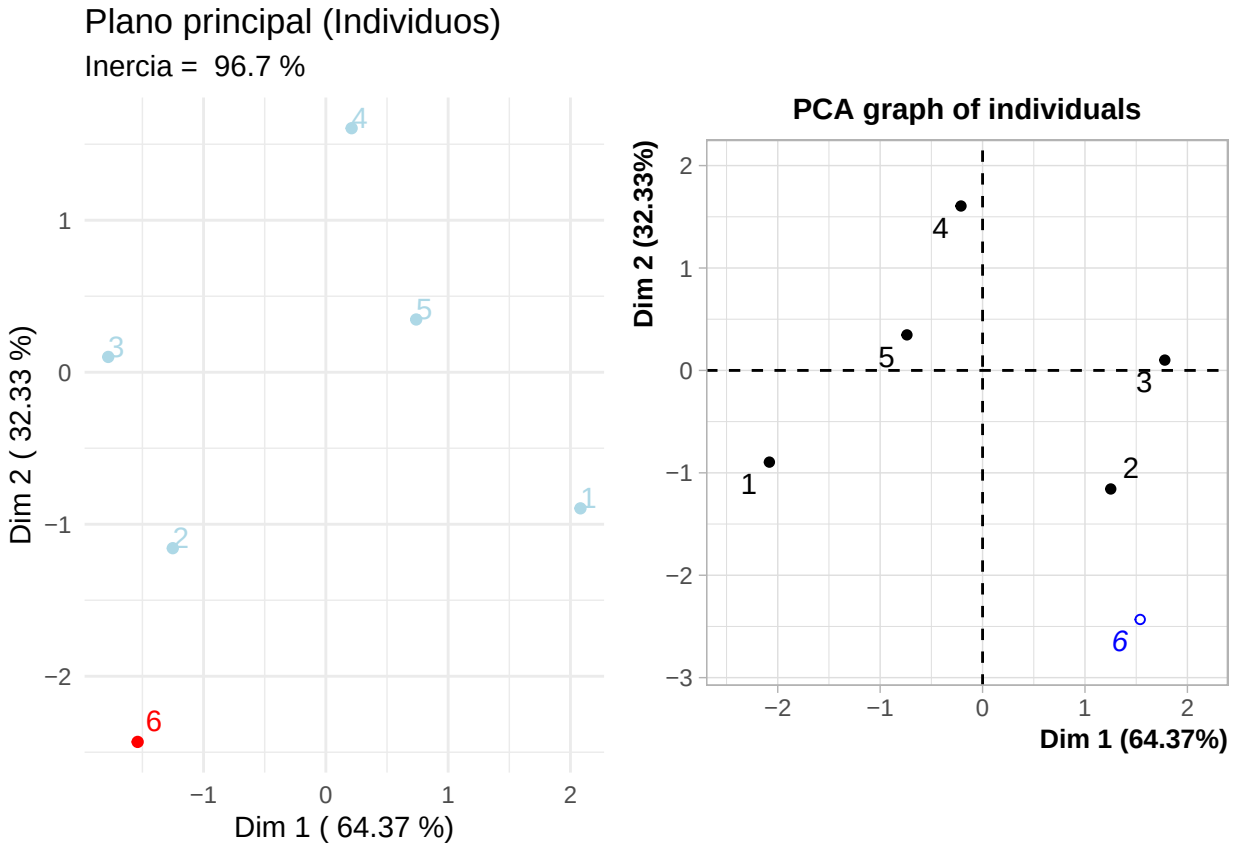
La función es la siguiente:

```
ind_sup_proyeccion <- function(fila, matriz) {

 #Obtener medias de las columnas de la matriz
  medias <-c()
  for(i in 1:ncol(matriz)) {
    medias[i] <- mean(matriz[,i])
  }

#desviaciones estándar poblacionales de la matriz
  sd <- c()
  n <- nrow(matriz)
  for(i in 1:ncol(matriz)) {
```

```
    sd[i] <- sqrt(((n-1)/n))*sd(matriz[,i])
  }

  #centramos y reducimos la fila y la matriz
  for(i in 1:length(fila)){
    fila[i] <- (fila[i]-medias[[i]])/sd[[i]]
  }

  matriz <- centrar_y_reducir(matriz)

  #Matriz de correlaciones
  correlaciones <- matriz_correlaciones_ind(matriz)

  #Vectores propios
  matriz_e <- eigen(correlaciones)
  V <- matriz_e$vectors

  #Se calcula las coordenadas
  C <- fila%*%V

  #Se gráfica esas coordenadas en el plano mediante la conversión de la fila en
  # un dataframe.
  C_data_ind <- as.data.frame(C)
  col_names <- paste("Dim", 1:ncol(C_data_ind))
  colnames(C_data_ind) <- col_names

  if(is.null(rownames(fila))){
    C_data_ind$individuo <- nrow(matriz) + 1
  }else {
    C_data_ind$individuo <- rownames(fila)
  }

  resultado <- plano_principal(matriz) +
    geom_point(data = C_data_ind, aes(x = `Dim 1`, y = `Dim 2`), color = "red")+
    geom_text(data = C_data_ind, aes(x = `Dim 1`, y = `Dim 2`),
              vjust = -0.5, hjust = -0.5, color = "red",
              label = C_data_ind$individuo )
  return(resultado)

}
```

Comparamos la proyección suplementaria del individuo en el plano inicial que da esta función con el resultado de FactoMiner. Para esto, tomamos como individuo suplementario a la fila número 6 de la matriz X.

```
fila <- X[6,]
coordenadasind_sup <- ind_sup_proyeccion(fila, X[-6,])

ACP_indsup <- PCA( X,ind.sup = 6, graph = F)
coordenadasind_sup_FM <-plot.PCA(ACP_indsup, choix = "ind")

grid.arrange(coordenadasind_sup, coordenadasind_sup_FM, ncol = 2)
```



Podemos observar que los gráficos se encuentran reflejados con respecto al eje "y". Además, el punto de color distinto, en cada gráfico, corresponde al individuo suplementario. También, es posible notar que al agregar la proyección suplementaria no se alteran las coordenadas de los individuos iniciales.

Ejercicio 5

Programe una función en R que reciba una columna (variable) de una matriz y calcule su proyección en suplementario en el círculo de correlaciones 2D programado en el punto2. Compare los resultados obtenidos con respecto a FactoMineR.

Primeramente, es necesario mencionar que la variable suplementaria se considera como una nueva columna que se le añade a la base de datos inicial. Por lo tanto, en caso de que la columna no tenga una etiqueta, se le coloca un identificador que corresponde al número de columna, en este caso, el último.

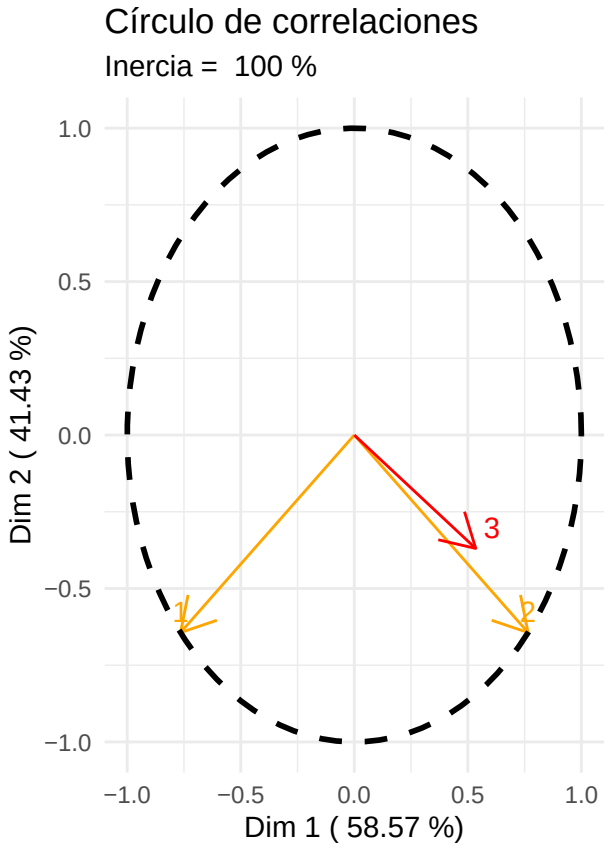
La función es la siguiente:

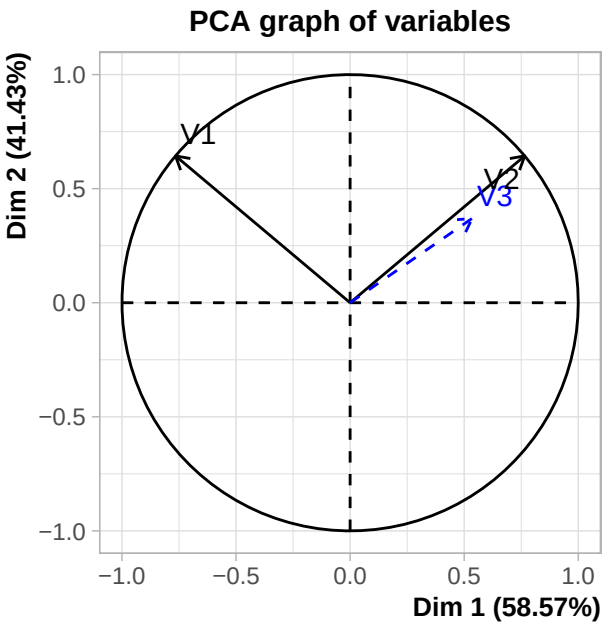
```
var_sup_proyeccion <- function(columna, matriz) {
  #se calcula la media y desviación estándar de la columna
  media <- mean(columna)
  n <- length(columna)
  sd <-sqrt(((n-1)/n))*sd(columna)

  #Se obtienen los componentes principales
  ACP_matriz <- ACP_funcion(matriz)
  C <- ACP_matriz$ind$coord

  #Se calculan las correlaciones de la columna con los componentes principales
```

```
  coordenada <- c()
  for(i in 1: ncol(matriz)){
    coordenada[i] <-cor(columna, C[,i])
  }

  #Se gráfica en el círculo de correlaciones
  circulo <- circulo_correlaciones(matriz)
  graf_circulo <- circulo$variables

  #Convertimos la matriz en un dataframe y ajustamos para el gráfico
  coordenada_var <- as.data.frame(t(coordenada))
  col_names <- paste("Dim", 1:ncol(coordenada_var))
  colnames(coordenada_var) <- col_names
  coordenada_var$`x origen` <- 0
  coordenada_var$`y origen` <- 0

  if(is.null(colnames(columna))){
   coordenada_var$variable <- ncol(matriz) + 1
  }else {
    coordenada_var$variable <- colnames(columna)
  }

  resultado <- graf_circulo +
    geom_segment(data = coordenada_var,
                 aes(x = `x origen`, y = `y origen`, xend = `Dim 1`,
                     yend = `Dim 2`),
    arrow = arrow(length = unit(0.2, "inches")), color = "red")+
    geom_text(data = coordenada_var, aes(x = `Dim 1`, y = `Dim 2`),
              vjust = -0.5, hjust = -0.5, color = "red",
              label = coordenada_var$variable )
  return(resultado)
}
```

Para comparar con FactoMiner, tomamos como variable suplementaria a la columna 3.

```
columna <- X[,3]
coordenadasvar_sup <- var_sup_proyeccion(columna, X[,-3])

ACP_varsup <- PCA( X,quanti.sup = 3, graph = F)
coordenadasvar_sup_FM <- plot.PCA(ACP_varsup, choix = "var")

grid.arrange(coordenadasvar_sup, coordenadasvar_sup_FM, ncol = 2)
```





Para el caso de la variable suplementaria, se puede ver que los gráficos se encuentran reflejados con respecto al eje x. También, el vector con distinto color, en cada gráfico, corresponde a la variable suplementaria. Además, es posible observar que al agregar la proyección suplementaria no se alteran las coordenadas de las variables iniciales.

Ejercicio 6

Programe en R un algoritmo para el ACP que en lugar de calcular la matriz de correlaciones R calcule H y con base en H realice los cálculos de lado de las variables y luego usando relaciones de dualidad realice todos los cálculos para la parte de los individuos. Verifique los resultados obtenidos con respecto a FactoMineR.

```
ACP_basado_en_H <- function(matriz_cyr){
  # Se crea una función para generar la matriz H.
  calcular_H <- function(matriz){
    # Se retorna la matriz de Varianzas-covarianzas.
    return((1/nrow(matriz))* matriz %*% t(matriz))
  }

  # Se corre la función y se genera la matriz H.
  H <- calcular_H(matriz_cyr)

  # Se calculan los valores y vectores propios de la matriz H.
  e <- eigen(H)
  val_propios_var <- e$values
```

```r
vec_propios_var <- e$vectors

# Dado que pude suceder que se obtengan valores propios cercanos a cero,
# entonces se procede a filtrar los valores y vectores propios.
val_propios_var <- val_propios_var[val_propios_var > 1e-10]
vec_propios_var <- vec_propios_var[,1:length(val_propios_var)]



# Se crea una función que calcule las coordenadas de las variables.
calcular_coordenadas <- function(matriz, vectores.propios){
  # Se obtiene la cantidad de filas y columnas que posee la matriz.
  m <- ncol(vectores.propios)
  n <- ncol(matriz)

  # Se crea una matriz en blanco la cual se rellenará con las coordenadas.
  coordenadas <- matrix(0,n,m)

  # Se rellena la matriz mediante un resultado obtenido a partir de la
  # relación de dualidad.
  for(i in 1:n){
    for(j in 1:m){
      x <- matriz_cyr[,i]
      v <- vec_propios_var[,j]

      coordenadas[i,j] <- t(x)%*%v/sqrt(t(x)%*%x%*%t(v)%*%v)
    }
  }

  return(coordenadas)
}

# Se calcula la matriz de coordenadas para las variables.
coordenadas_var <- calcular_coordenadas(matriz_cyr, vec_propios_var)

# Una vez que se cuenta con matriz de coordenadas, se procede a elevar al
# cuadrado dicha matriz para obtener la matriz de cosenos cuadrados. Además se
# compara el resultado con el obtenido con FactoMineR.
cos2_var <- (coordenadas_var)^2

# Se crea una función para calcular la contribución de las variables.
calcular_contribuciones <- function(matriz){
  m <- ncol(matriz)
  n <- nrow(matriz)

  # Se genera una matriz vacía a rellenar con las contribuciones.
  resultado <- matrix(0,n,m)

  for(i in 1:m){
    for(j in 1:n){
      resultado[j,i] <- matriz[j,i]/sum(matriz[,i])*100
    }
  }
```

```
  return(resultado)
}

# Se calculan las contribuciones.
contribuciones_var <- calcular_contribuciones(cos2_var)

# Ahora que se cuenta con toda la información útil sobre las variables, se
# se procede a crear una función para calcular los vectores propios de los
# individuos.
calcular_vec_propios_ind <- function(matriz, vec_p_v, val_p_v){
  n <- ncol(matriz)
  f <- nrow(matriz)
  m <- ncol(vec_p_v)

  # Se genera una matriz vacía a rellenar con los vectores propios de los
  # individuos.
  resultado <- matrix(0,n,m)

  for(i in 1:m){
    resultado[,i] <- (t(matriz)%*%vec_p_v[,i])/sqrt(f*val_p_v[i])
  }

  return(resultado)
}

# Se calculan los vectores propios de los individuos.
vec_propios_ind <- calcular_vec_propios_ind(matriz_cyr, vec_propios_var,
                                            val_propios_var)

# Se calculan las coordenadas de los individuos.
coordenadas_ind <- matriz_cyr%*%vec_propios_ind

# Se crea la funcion para el calculo de los cosenos cuadrados.
calcular_cos2_ind <- function(C, matriz) {
  n <-nrow(matriz)
  m <-ncol(matriz)
  f <- ncol(C)


  resultado <- matrix(0, n,f)

  for(i in 1: n){
    suma <- 0
    for(j in 1: m){
      suma <- suma + matriz[i,j]^2
    }
    for(r in 1: f){
      resultado[i,r] <- (C[i,r]^2)/suma
    }
  }
  return(resultado)
}
```

```
  # Se calcula la matriz de cosenos cuadrados.
  cos2_ind <- calcular_cos2_ind(coordenadas_ind, matriz_cyr)

  # Se calcula la matriz de contribuciones.
  contribuciones_ind <- contribucion_ind(coordenadas_ind, val_propios_var)

  # Se calcula la inercia.
  inercia <- calcular_inercias(val_propios_var)

  resultado <- list(
    eigen = list(values = val_propios_var, vectors = vec_propios_ind),
    var = list(coord = coordenadas_var, cos2 = cos2_var,
               contrib = contribuciones_var),
    ind = list(coord = coordenadas_ind, cos2 = cos2_ind,
               contrib = contribuciones_ind)
  )

  return(resultado)
}
```

Finalmente se corre todo el algoritmo y se compara los resultados con los obtenidos por FactoMineR.

```
X_cyr <- centrar_y_reducir(X)

ACP_a_mano <- ACP_basado_en_H(X_cyr)
ACP_con_FM <- PCA(X_cyr, graph = FALSE)

# Valores propios,
ACP_a_mano$eigen$values
```

```
## [1] 1.7278743 0.9191845 0.3529412
```

```
ACP_con_FM$eig
```

```
##        eigenvalue percentage of variance cumulative percentage of variance
## comp 1  1.7278743              57.59581                           57.59581
## comp 2  0.9191845              30.63948                           88.23529
## comp 3  0.3529412              11.76471                          100.00000
```

```
# Coordenadas de las variables.
ACP_a_mano$var$coord
```

```
##            [,1]       [,2]          [,3]
## [1,]  0.4155396 -0.9095751  9.064933e-17
## [2,] -0.8818166 -0.2143101  4.200840e-01
## [3,] -0.8818166 -0.2143101 -4.200840e-01
```

```
ACP_con_FM$var$coord
```

```
##         Dim.1     Dim.2     Dim.3
## V1 -0.4155396 0.9095751  0.000000
## V2  0.8818166 0.2143101  0.420084
## V3  0.8818166 0.2143101 -0.420084
```

```r
# Coseno cuadrado de las variables.
ACP_a_mano$var$cos2
```

```
##           [,1]       [,2]         [,3]
## [1,] 0.1726732 0.82732684 8.217301e-33
## [2,] 0.7776006 0.04592883 1.764706e-01
## [3,] 0.7776006 0.04592883 1.764706e-01
```

```r
ACP_con_FM$var$cos2
```

```
##        Dim.1      Dim.2     Dim.3
## V1 0.1726732 0.82732684 0.0000000
## V2 0.7776006 0.04592883 0.1764706
## V3 0.7776006 0.04592883 0.1764706
```

```r
# Contribuciones de las variables.
ACP_a_mano$var$contrib
```

```
##           [,1]      [,2]         [,3]
## [1,]  9.993387 90.006613 2.328235e-30
## [2,] 45.003307  4.996693 5.000000e+01
## [3,] 45.003307  4.996693 5.000000e+01
```

```r
ACP_con_FM$var$contrib
```

```
##        Dim.1     Dim.2 Dim.3
## V1  9.993387 90.006613     0
## V2 45.003307  4.996693    50
## V3 45.003307  4.996693    50
```

```r
# Coordenadas de los individuos.
ACP_a_mano$ind$coord
```

```
##            [,1]       [,2]       [,3]
## [1,]  2.4480754  0.1702518  0.5940885
## [2,] -0.7572077  0.3931637  0.5940885
## [3,] -1.6908677 -0.5634155  0.5940885
## [4,] -0.1764523 -1.3497428 -0.5940885
## [5,]  0.7572077 -0.3931637 -0.5940885
## [6,] -0.5807554  1.7429065 -0.5940885
```

```r
ACP_con_FM$ind$coord
```

```
##        Dim.1      Dim.2      Dim.3
## 1 -2.4480754 -0.1702518  0.5940885
## 2  0.7572077 -0.3931637  0.5940885
## 3  1.6908677  0.5634155  0.5940885
## 4  0.1764523  1.3497428 -0.5940885
## 5 -0.7572077  0.3931637 -0.5940885
## 6  0.5807554 -1.7429065 -0.5940885
```

```r
# Coseno cuadrado de los individuos.
ACP_a_mano$ind$cos2
```

```
##            [,1]        [,2]       [,3]
## [1,] 0.94008991 0.004546772 0.05536332
## [2,] 0.53045874 0.143010648 0.32653061
## [3,] 0.81005952 0.089940481 0.10000000
## [4,] 0.01411472 0.825885276 0.16000000
## [5,] 0.53045874 0.143010648 0.32653061
## [6,] 0.09047267 0.814852770 0.09467456
```

```r
ACP_con_FM$ind$cos2
```

```
##        Dim.1       Dim.2      Dim.3
## 1 0.94008991 0.004546772 0.05536332
## 2 0.53045874 0.143010648 0.32653061
## 3 0.81005952 0.089940481 0.10000000
## 4 0.01411472 0.825885276 0.16000000
## 5 0.53045874 0.143010648 0.32653061
## 6 0.09047267 0.814852770 0.09467456
```

```r
# Contribuciones de los individuos.
ACP_a_mano$ind$contrib
```

```
##            [,1]       [,2]     [,3]
## [1,] 57.8077648  0.5255686 16.66667
## [2,]  5.5305285  2.8028049 16.66667
## [3,] 27.5775612  5.7557722 16.66667
## [4,]  0.3003249 33.0330084 16.66667
## [5,]  5.5305285  2.8028049 16.66667
## [6,]  3.2532922 55.0800411 16.66667
```

```r
ACP_con_FM$ind$contrib
```

```
##        Dim.1      Dim.2    Dim.3
## 1 57.8077648  0.5255686 16.66667
## 2  5.5305285  2.8028049 16.66667
## 3 27.5775612  5.7557722 16.66667
## 4  0.3003249 33.0330084 16.66667
## 5  5.5305285  2.8028049 16.66667
## 6  3.2532922 55.0800411 16.66667
```

Ejercicio 7

Programe en R un algoritmo óptimo para el ACP que tome en cuenta cual matriz sea más pequeña R o H.

```r
ACP <- function(X){
  if(nrow(X) >= ncol(X)){
    return(ACP_funcion(X))
  } else{
    X_cyr <- centrar_y_reducir(X)
    return(ACP_basado_en_H(X_cyr))
  }
}
```

Ejercicio 8

Verifique todo lo programado en los puntos anteriores con el ejemplo estudiantes.csv y con los datos del ejercicio 1 de la tarea anterior.

Base de datos de estudiantes

Inciso 2

```r
ACP_estudiantes <- ACP_funcion(estudiantes_datos)
ACP_estudiantes
```

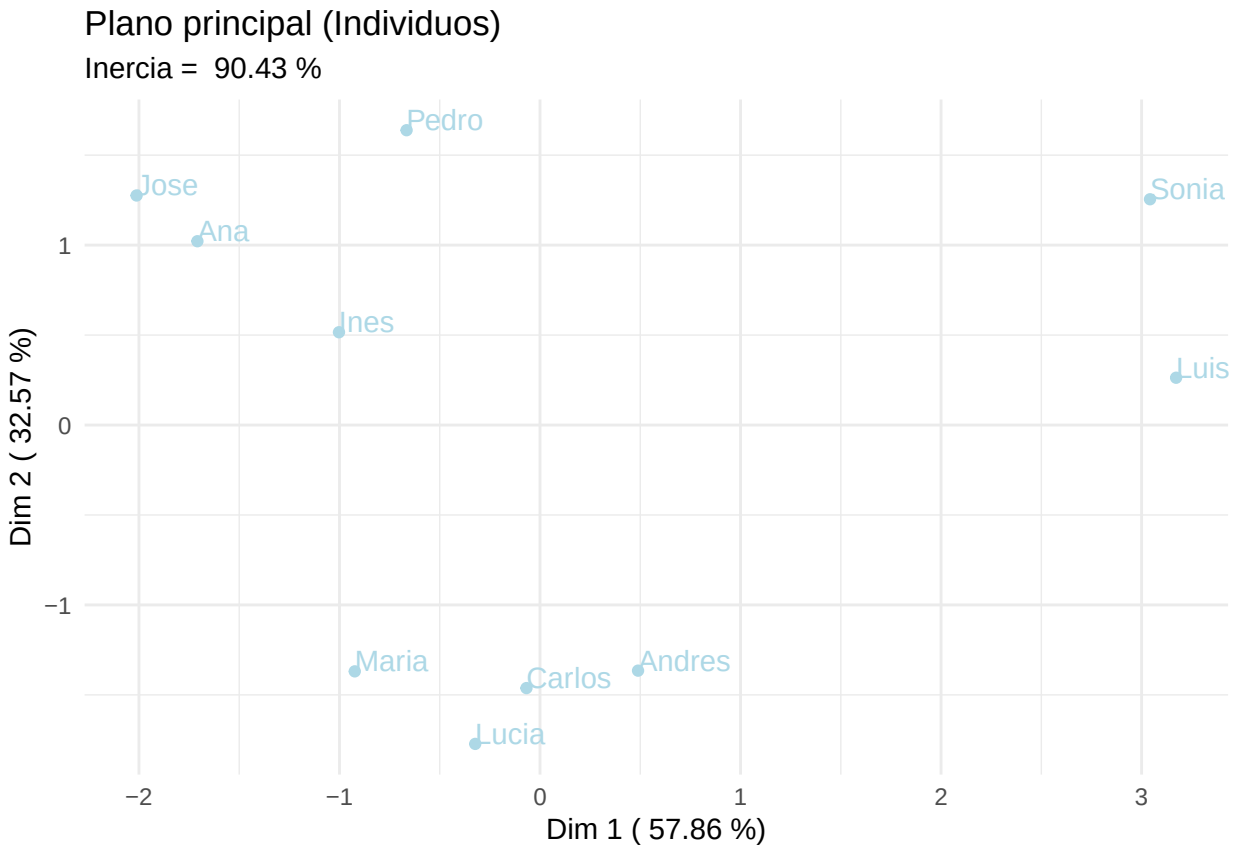
```
## $valores_propios
## [1] 2.893249673 1.628650425 0.346596049 0.122612460 0.008891393
##
## $ind
## $ind$coord
##               [,1]       [,2]        [,3]        [,4]         [,5]
## Lucia  -0.32306263 -1.7725245  1.19880074 -0.05501532  0.003633384
## Pedro  -0.66544057  1.6387021  0.14547628 -0.02306468 -0.123377296
## Ines   -1.00254705  0.5156925  0.62888764  0.51644351  0.142875824
## Luis    3.17209481  0.2627820 -0.38196027  0.67777691 -0.062503554
## Andres  0.48886797 -1.3654021 -0.83523570 -0.15579197  0.123367255
## Ana    -1.70863322  1.0217004 -0.12707707  0.06683295  0.025291503
## Carlos -0.06758577 -1.4623364 -0.50624044 -0.11792847  0.013123980
## Jose   -2.01185516  1.2758646 -0.54215002 -0.19778670  0.017434170
## Sonia   3.04203029  1.2548807  0.44882861 -0.63999876  0.037884840
## Maria  -0.92386867 -1.3693593 -0.02932977 -0.07146746 -0.177730107
##
## $ind$cos2
##              [,1]        [,2]        [,3]         [,4]         [,5]
##  [1,] 0.022270827 0.670420670 0.306659839 0.0006458478 2.816992e-06
##  [2,] 0.139905502 0.848430539 0.006686527 0.0001680781 4.809354e-03
##  [3,] 0.514468899 0.136122895 0.202439714 0.1365196756 1.044882e-02
##  [4,] 0.936851990 0.006429392 0.013583605 0.0427712757 3.637375e-04
##  [5,] 0.084139511 0.656353715 0.245603703 0.0085448999 5.358172e-03
##  [6,] 0.732686110 0.261979570 0.004052795 0.0011209894 1.605349e-04
##  [7,] 0.001892733 0.886081139 0.106192189 0.0057625700 7.136907e-05
##  [8,] 0.673612108 0.270910359 0.048916504 0.0065104446 5.058468e-05
##  [9,] 0.808829929 0.137636943 0.017607237 0.0358004434 1.254472e-04
```

```
## [10,] 0.308554271 0.677869212 0.000310977 0.0018464085 1.141913e-02
##
## $ind$contrib
##              [,1]       [,2]        [,3]        [,4]        [,5]
##  [1,]  0.36073437 19.2910834 41.46392357  0.24684974  0.01484748
##  [2,]  1.53049754 16.4881591  0.61060555  0.04338706 17.11987788
##  [3,]  3.47395038  1.6328779 11.41096846 21.75259335 22.95871968
##  [4,] 34.77814436  0.4239976  4.20932799 37.46613853  4.39379307
##  [5,]  0.82603273 11.4470414 20.12771563  1.97950024 17.11709152
##  [6,] 10.09047896  6.4094282  0.46591936  0.36428947  0.71941493
##  [7,]  0.01578791 13.1300601  7.39418080  1.13423412  0.19371414
##  [8,] 13.98967133  9.9949649  8.48038057  3.19050613  0.34184774
##  [9,] 31.98461714  9.6688984  5.81215853 33.40593699  1.61421395
## [10,]  2.95008527 11.5134890  0.02481953  0.41656436 35.52647960
##
##
## $var
## $var$coord
##            [,1]       [,2]        [,3]        [,4]        [,5]
## [1,] -0.8957980  0.3452036  0.25797931 -0.09146818  0.05882803
## [2,] -0.7227976  0.6483946  0.02384033  0.23587773 -0.03068234
## [3,] -0.6108931 -0.7173206  0.33102532 -0.02454152 -0.04561456
## [4,] -0.5999227 -0.7484701 -0.23206345  0.15639747  0.03964443
## [5,]  0.9139265 -0.1196373  0.34065108  0.18315368  0.02892890
##
## $var$cos2
##           [,1]       [,2]         [,3]         [,4]         [,5]
## [1,] 0.8024540 0.11916550 0.0665533242 0.0083664287 0.0034607374
## [2,] 0.5224364 0.42041555 0.0005683612 0.0556383052 0.0009414059
## [3,] 0.3731904 0.51454884 0.1095777630 0.0006022863 0.0020806881
## [4,] 0.3599073 0.56020745 0.0538534429 0.0244601695 0.0015716811
## [5,] 0.8352616 0.01431309 0.1160431572 0.0335452699 0.0008368811
##
## $var$contrib
##          [,1]       [,2]       [,3]       [,4]     [,5]
## [1,] 27.73539  7.3168250 19.2019858  6.8234735 38.92233
## [2,] 18.05708 25.8137375  0.1639838 45.3773665 10.58783
## [3,] 12.89866 31.5935718 31.6154103  0.4912113 23.40115
## [4,] 12.43955 34.3970346 15.5378121 19.9491712 17.67643
## [5,] 28.86932  0.8788311 33.4808079 27.3587774  9.41226
##
##
## $inercias
## [1] 57.8649935 32.5730085  6.9319210  2.4522492  0.1778279
```

Inciso 3

Plano principal

```
plano_principal(estudiantes_datos)
```



Círculo de correlaciones

```
circulo_correlaciones(estudiantes_datos)$variables
```

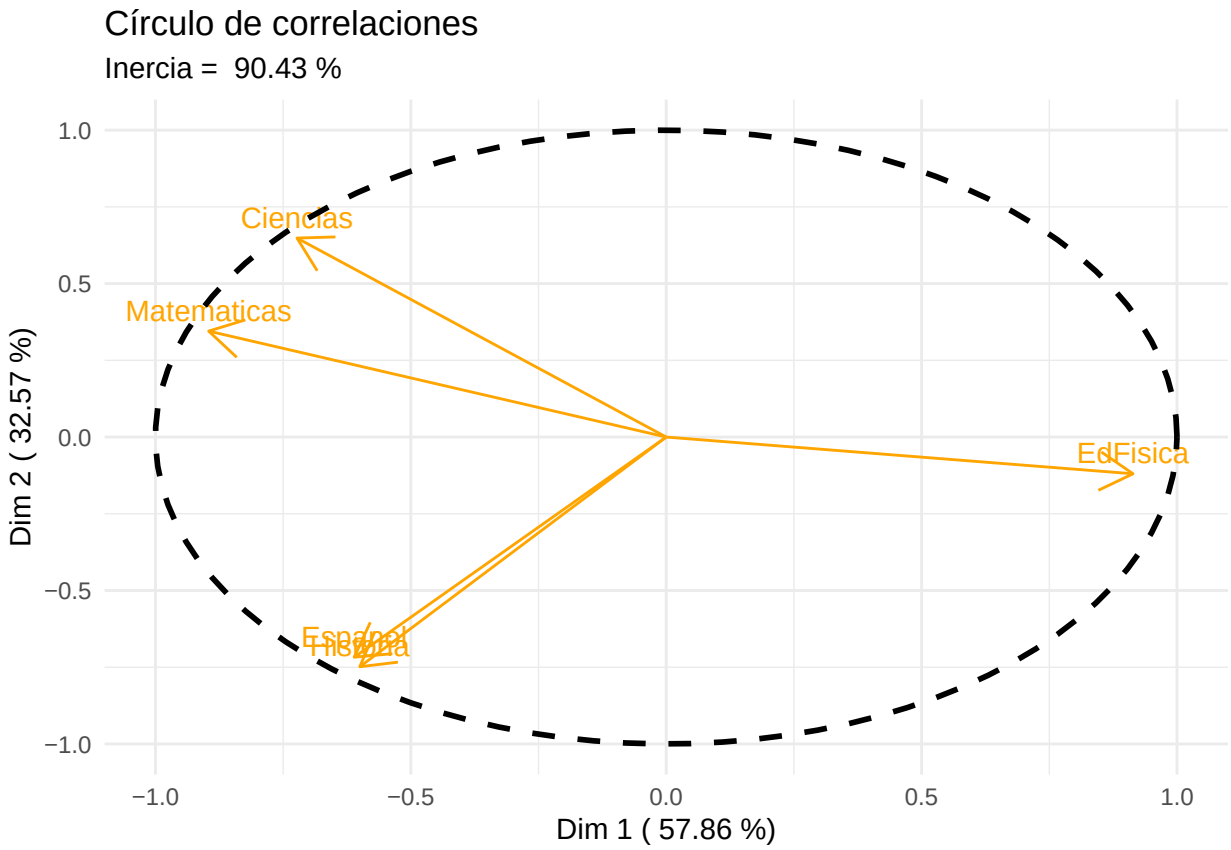
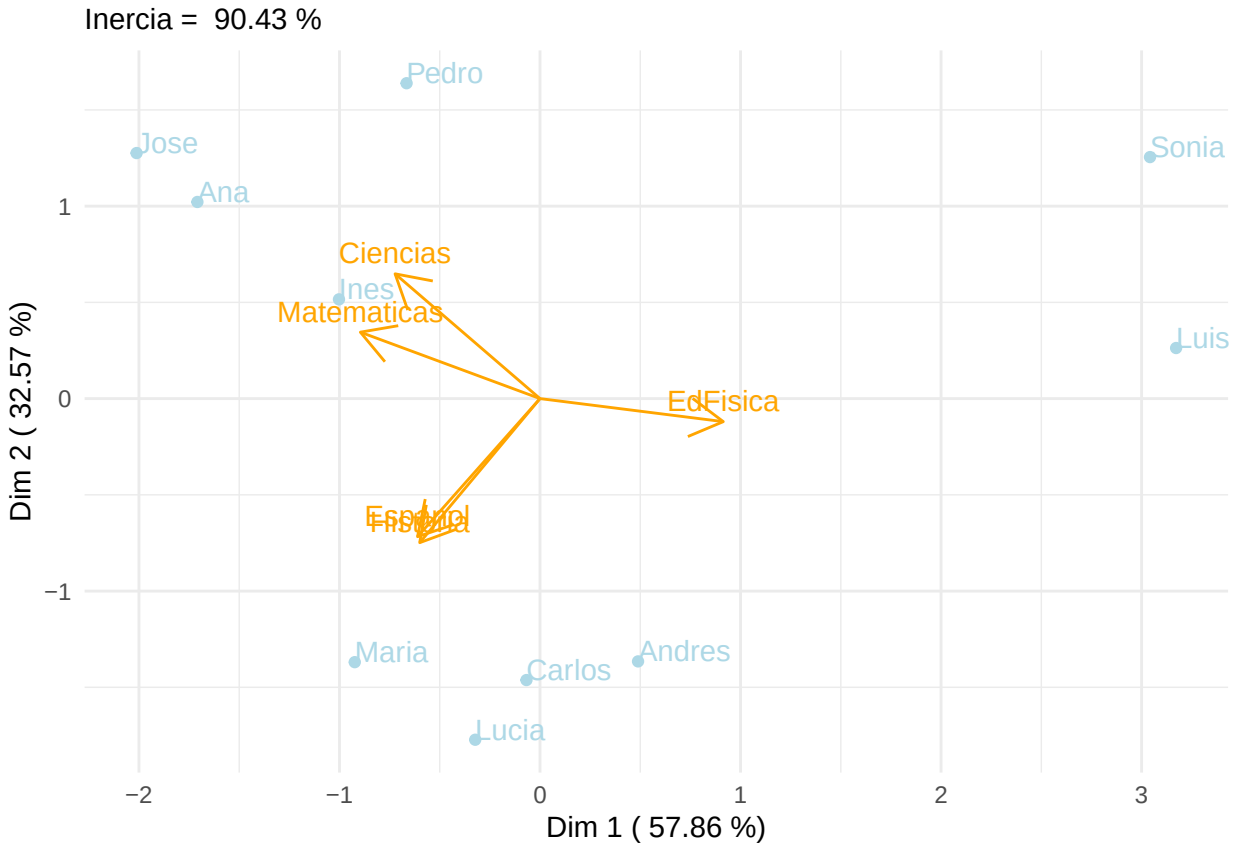


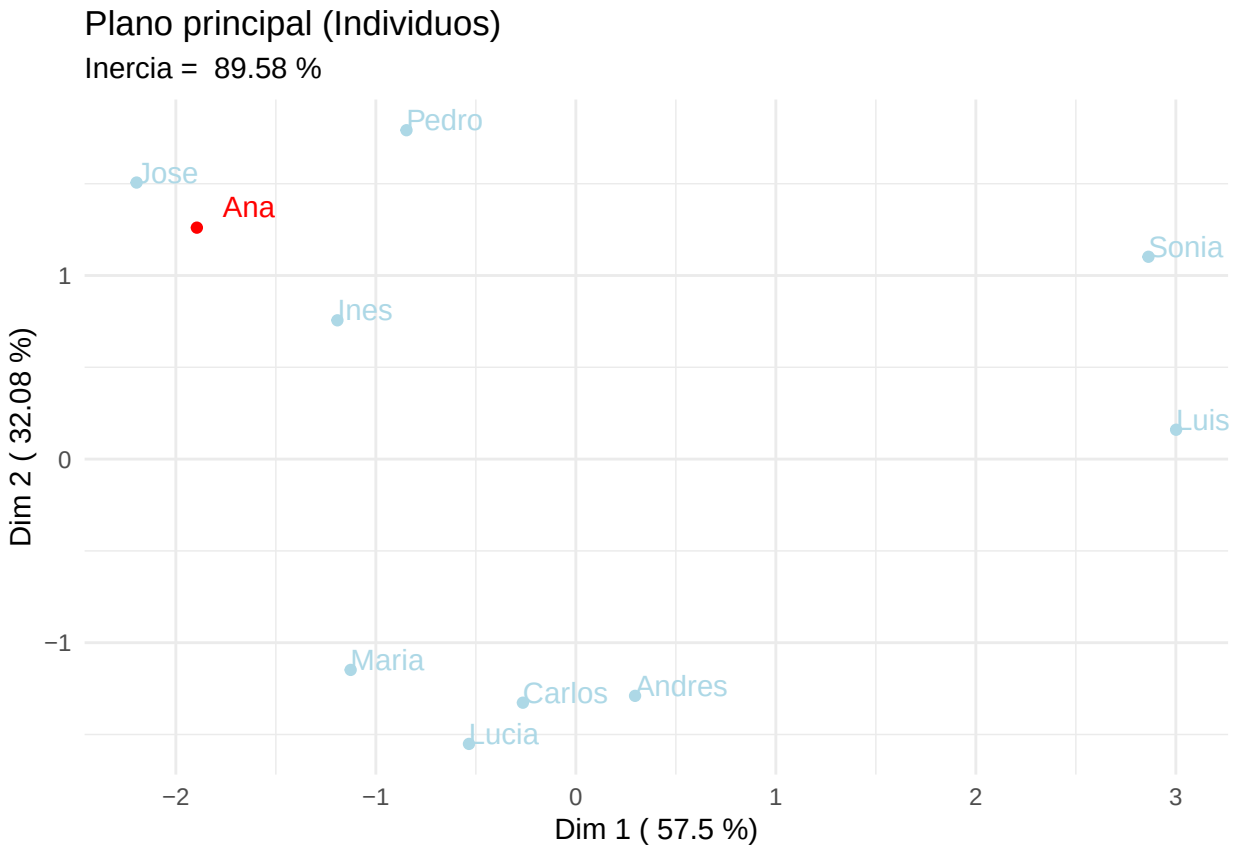
Gráfico Dual

```
grafico_dual(plano_principal(estudiantes_datos),
             circulo_correlaciones(estudiantes_datos), estudiantes_datos)
```



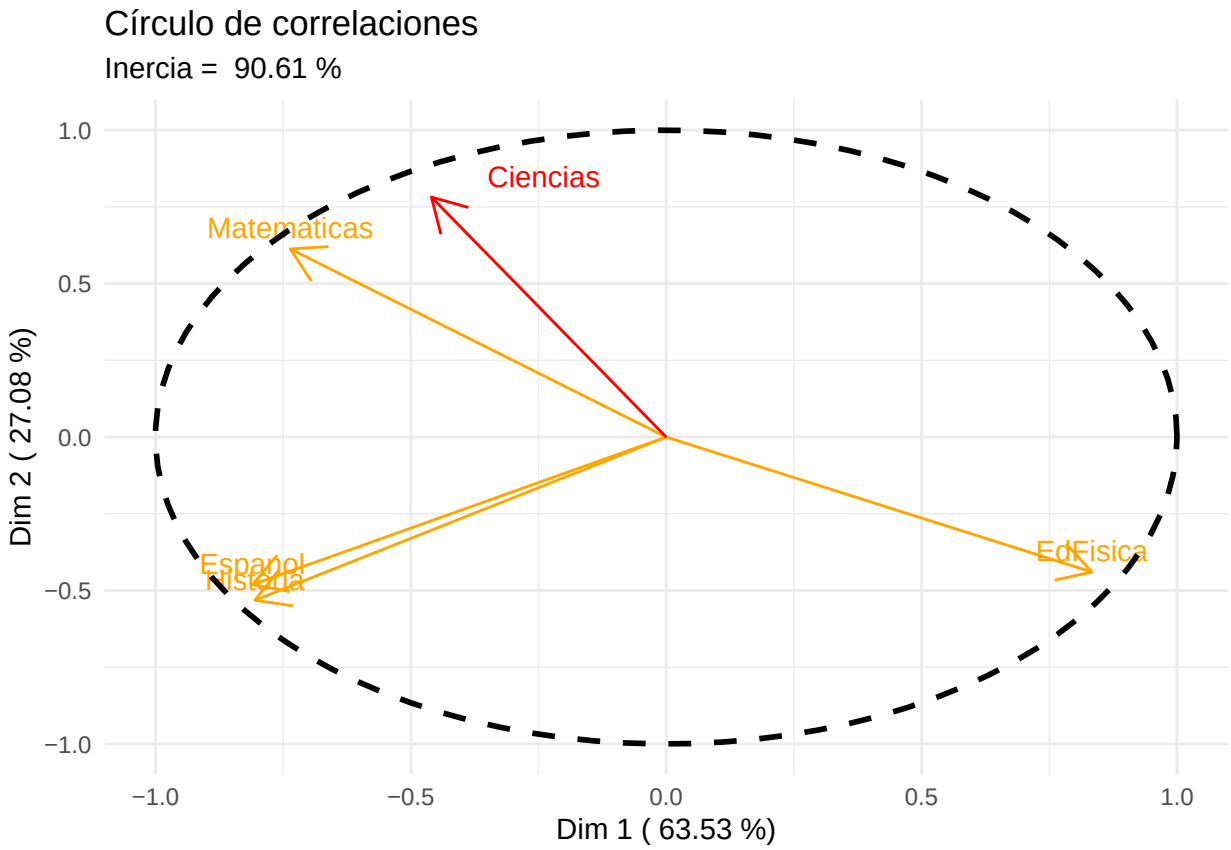
Inciso 4

```
fila_sup_estudiantes <- as.matrix(estudiantes_datos[6,])
colnames(fila_sup_estudiantes) <- (rownames(estudiantes_datos))[6]
fila_sup_estudiantes <- t(fila_sup_estudiantes)

coordenadasind_sup_estudiantes <- ind_sup_proyeccion(fila_sup_estudiantes,
                                                     estudiantes_datos[-6,])
coordenadasind_sup_estudiantes
```



Inciso 5

```
columna_sup_estudiantes <- as.matrix(estudiantes_datos[,2])
colnames(columna_sup_estudiantes) <- (colnames(estudiantes_datos))[2]

coordenadasvar_sup_estudiantes <- var_sup_proyeccion(columna_sup_estudiantes,
                                                     estudiantes_datos[,-2])
coordenadasvar_sup_estudiantes
```



Inciso 6

```
ACP_estudiantes_H <- ACP_basado_en_H(centrar_y_reducir(estudiantes_datos))
ACP_estudiantes_H
```

```
## $eigen
## $eigen$values
## [1] 2.893249673 1.628650425 0.346596049 0.122612460 0.008891393
##
## $eigen$vectors
##            [,1]        [,2]        [,3]        [,4]       [,5]
## [1,]  0.5266440 -0.27049630  0.43820071 -0.26121779 -0.6238776
## [2,]  0.4249362 -0.50807221  0.04049491  0.67362724  0.3253895
## [3,]  0.3591470  0.56208159  0.56227583 -0.07008647  0.4837473
## [4,]  0.3526975  0.58648985 -0.39418032  0.44664495 -0.4204335
## [5,] -0.5373018  0.09374599  0.57862603  0.52305619 -0.3067941
##
##
## $var
## $var$coord
##            [,1]       [,2]        [,3]        [,4]        [,5]
## [1,]  0.8957980 -0.3452036  0.25797931 -0.09146818 -0.05882803
## [2,]  0.7227976 -0.6483946  0.02384033  0.23587773  0.03068234
## [3,]  0.6108931  0.7173206  0.33102532 -0.02454152  0.04561456
```

```
## [4,]  0.5999227  0.7484701 -0.23206345  0.15639747 -0.03964443
## [5,] -0.9139265  0.1196373  0.34065108  0.18315368 -0.02892890
##
## $var$cos2
##           [,1]       [,2]         [,3]         [,4]         [,5]
## [1,] 0.8024540 0.11916550 0.0665533242 0.0083664287 0.0034607374
## [2,] 0.5224364 0.42041555 0.0005683612 0.0556383052 0.0009414059
## [3,] 0.3731904 0.51454884 0.1095777630 0.0006022863 0.0020806881
## [4,] 0.3599073 0.56020745 0.0538534429 0.0244601695 0.0015716811
## [5,] 0.8352616 0.01431309 0.1160431572 0.0335452699 0.0008368811
##
## $var$contrib
##           [,1]       [,2]       [,3]       [,4]     [,5]
## [1,] 27.73539  7.3168250 19.2019858  6.8234735 38.92233
## [2,] 18.05708 25.8137375  0.1639838 45.3773665 10.58783
## [3,] 12.89866 31.5935718 31.6154103  0.4912113 23.40115
## [4,] 12.43955 34.3970346 15.5378121 19.9491712 17.67643
## [5,] 28.86932  0.8788311 33.4808079 27.3587774  9.41226
##
##
## $ind
## $ind$coord
##               [,1]       [,2]        [,3]        [,4]         [,5]
## Lucia   0.32306263  1.7725245  1.19880074 -0.05501532 -0.003633384
## Pedro   0.66544057 -1.6387021  0.14547628 -0.02306468  0.123377296
## Ines    1.00254705 -0.5156925  0.62888764  0.51644351 -0.142875824
## Luis   -3.17209481 -0.2627820 -0.38196027  0.67777691  0.062503554
## Andres -0.48886797  1.3654021 -0.83523570 -0.15579197 -0.123367255
## Ana     1.70863322 -1.0217004 -0.12707707  0.06683295 -0.025291503
## Carlos  0.06758577  1.4623364 -0.50624044 -0.11792847 -0.013123980
## Jose    2.01185516 -1.2758646 -0.54215002 -0.19778670 -0.017434170
## Sonia  -3.04203029 -1.2548807  0.44882861 -0.63999876 -0.037884840
## Maria   0.92386867  1.3693593 -0.02932977 -0.07146746  0.177730107
##
## $ind$cos2
##              [,1]        [,2]        [,3]         [,4]         [,5]
##  [1,] 0.022270827 0.670420670 0.306659839 0.0006458478 2.816992e-06
##  [2,] 0.139905502 0.848430539 0.006686527 0.0001680781 4.809354e-03
##  [3,] 0.514468899 0.136122895 0.202439714 0.1365196756 1.044882e-02
##  [4,] 0.936851990 0.006429392 0.013583605 0.0427712757 3.637375e-04
##  [5,] 0.084139511 0.656353715 0.245603703 0.0085448999 5.358172e-03
##  [6,] 0.732686110 0.261979570 0.004052795 0.0011209894 1.605349e-04
##  [7,] 0.001892733 0.886081139 0.106192189 0.0057625700 7.136907e-05
##  [8,] 0.673612108 0.270910359 0.048916504 0.0065104446 5.058468e-05
##  [9,] 0.808829929 0.137636943 0.017607237 0.0358004434 1.254472e-04
## [10,] 0.308554271 0.677869212 0.000310977 0.0018464085 1.141913e-02
##
## $ind$contrib
##              [,1]       [,2]        [,3]        [,4]        [,5]
##  [1,]  0.36073437 19.2910834 41.46392357  0.24684974  0.01484748
##  [2,]  1.53049754 16.4881591  0.61060555  0.04338706 17.11987788
##  [3,]  3.47395038  1.6328779 11.41096846 21.75259335 22.95871968
##  [4,] 34.77814436  0.4239976  4.20932799 37.46613853  4.39379307
##  [5,]  0.82603273 11.4470414 20.12771563  1.97950024 17.11709152
```

```
##  [6,] 10.09047896  6.4094282  0.46591936  0.36428947  0.71941493
##  [7,]  0.01578791 13.1300601  7.39418080  1.13423412  0.19371414
##  [8,] 13.98967133  9.9949649  8.48038057  3.19050613  0.34184774
##  [9,] 31.98461714  9.6688984  5.81215853 33.40593699  1.61421395
## [10,]  2.95008527 11.5134890  0.02481953  0.41656436 35.52647960
```

Inciso 7

```
ACP_estudiantes_optimo <- ACP(estudiantes_datos)
ACP_estudiantes_optimo
```

```
## $valores_propios
## [1] 2.893249673 1.628650425 0.346596049 0.122612460 0.008891393
##
## $ind
## $ind$coord
##               [,1]       [,2]        [,3]        [,4]         [,5]
## Lucia  -0.32306263 -1.7725245  1.19880074 -0.05501532  0.003633384
## Pedro  -0.66544057  1.6387021  0.14547628 -0.02306468 -0.123377296
## Ines   -1.00254705  0.5156925  0.62888764  0.51644351  0.142875824
## Luis    3.17209481  0.2627820 -0.38196027  0.67777691 -0.062503554
## Andres  0.48886797 -1.3654021 -0.83523570 -0.15579197  0.123367255
## Ana    -1.70863322  1.0217004 -0.12707707  0.06683295  0.025291503
## Carlos -0.06758577 -1.4623364 -0.50624044 -0.11792847  0.013123980
## Jose   -2.01185516  1.2758646 -0.54215002 -0.19778670  0.017434170
## Sonia   3.04203029  1.2548807  0.44882861 -0.63999876  0.037884840
## Maria  -0.92386867 -1.3693593 -0.02932977 -0.07146746 -0.177730107
##
## $ind$cos2
##              [,1]        [,2]        [,3]         [,4]         [,5]
##  [1,] 0.022270827 0.670420670 0.306659839 0.0006458478 2.816992e-06
##  [2,] 0.139905502 0.848430539 0.006686527 0.0001680781 4.809354e-03
##  [3,] 0.514468899 0.136122895 0.202439714 0.1365196756 1.044882e-02
##  [4,] 0.936851990 0.006429392 0.013583605 0.0427712757 3.637375e-04
##  [5,] 0.084139511 0.656353715 0.245603703 0.0085448999 5.358172e-03
##  [6,] 0.732686110 0.261979570 0.004052795 0.0011209894 1.605349e-04
##  [7,] 0.001892733 0.886081139 0.106192189 0.0057625700 7.136907e-05
##  [8,] 0.673612108 0.270910359 0.048916504 0.0065104446 5.058468e-05
##  [9,] 0.808829929 0.137636943 0.017607237 0.0358004434 1.254472e-04
## [10,] 0.308554271 0.677869212 0.000310977 0.0018464085 1.141913e-02
##
## $ind$contrib
##              [,1]       [,2]        [,3]        [,4]        [,5]
##  [1,]  0.36073437 19.2910834 41.46392357  0.24684974  0.01484748
##  [2,]  1.53049754 16.4881591  0.61060555  0.04338706 17.11987788
##  [3,]  3.47395038  1.6328779 11.41096846 21.75259335 22.95871968
##  [4,] 34.77814436  0.4239976  4.20932799 37.46613853  4.39379307
##  [5,]  0.82603273 11.4470414 20.12771563  1.97950024 17.11709152
##  [6,] 10.09047896  6.4094282  0.46591936  0.36428947  0.71941493
##  [7,]  0.01578791 13.1300601  7.39418080  1.13423412  0.19371414
##  [8,] 13.98967133  9.9949649  8.48038057  3.19050613  0.34184774
##  [9,] 31.98461714  9.6688984  5.81215853 33.40593699  1.61421395
```

```
## [10,]  2.95008527 11.5134890  0.02481953  0.41656436 35.52647960
##
##
## $var
## $var$coord
##            [,1]       [,2]        [,3]        [,4]        [,5]
## [1,] -0.8957980  0.3452036  0.25797931 -0.09146818  0.05882803
## [2,] -0.7227976  0.6483946  0.02384033  0.23587773 -0.03068234
## [3,] -0.6108931 -0.7173206  0.33102532 -0.02454152 -0.04561456
## [4,] -0.5999227 -0.7484701 -0.23206345  0.15639747  0.03964443
## [5,]  0.9139265 -0.1196373  0.34065108  0.18315368  0.02892890
##
## $var$cos2
##           [,1]       [,2]         [,3]         [,4]         [,5]
## [1,] 0.8024540 0.11916550 0.0665533242 0.0083664287 0.0034607374
## [2,] 0.5224364 0.42041555 0.0005683612 0.0556383052 0.0009414059
## [3,] 0.3731904 0.51454884 0.1095777630 0.0006022863 0.0020806881
## [4,] 0.3599073 0.56020745 0.0538534429 0.0244601695 0.0015716811
## [5,] 0.8352616 0.01431309 0.1160431572 0.0335452699 0.0008368811
##
## $var$contrib
##          [,1]       [,2]       [,3]       [,4]     [,5]
## [1,] 27.73539  7.3168250 19.2019858  6.8234735 38.92233
## [2,] 18.05708 25.8137375  0.1639838 45.3773665 10.58783
## [3,] 12.89866 31.5935718 31.6154103  0.4912113 23.40115
## [4,] 12.43955 34.3970346 15.5378121 19.9491712 17.67643
## [5,] 28.86932  0.8788311 33.4808079 27.3587774  9.41226
##
##
## $inercias
## [1] 57.8649935 32.5730085  6.9319210  2.4522492  0.1778279
```

Base de datos beans

```
ACP_beans <- ACP_funcion(beans_datos)
head(ACP_beans$ind$coord[,1:3],15)
```

```
##           [,1]     [,2]       [,3]
##  [1,] 5.065900 2.815875 -0.8358610
##  [2,] 4.916324 2.666202 -0.4053238
##  [3,] 5.350048 3.392308 -0.9013560
##  [4,] 5.369507 3.387817 -0.8869312
##  [5,] 4.809762 2.598609 -0.2132709
##  [6,] 4.235387 2.410951 -2.3307945
##  [7,] 4.924846 3.062623 -1.2018822
##  [8,] 4.234363 2.203307 -1.1495689
##  [9,] 5.607403 3.809293 -0.6862977
## [10,] 5.298258 3.582976 -1.3085764
## [11,] 3.849968 1.614777 -0.3128586
## [12,] 5.064607 3.262641 -0.9991558
## [13,] 4.136009 2.133879 -1.1644241
## [14,] 4.878775 2.868413 -0.3511886
```

```
## [15,] 5.131171 3.357005 -0.6760195
```

```
head(ACP_beans$ind$cos2[,1:3],15)
```

```
##            [,1]      [,2]        [,3]
##  [1,] 0.6687338 0.2066174 0.018205731
##  [2,] 0.6878382 0.2022979 0.004675300
##  [3,] 0.6277844 0.2523979 0.017819190
##  [4,] 0.6296584 0.2506549 0.017179720
##  [5,] 0.6804289 0.1986176 0.001337824
##  [6,] 0.4974925 0.1612045 0.150663546
##  [7,] 0.6128428 0.2370013 0.036499555
##  [8,] 0.6362952 0.1722791 0.046897806
##  [9,] 0.6069936 0.2801235 0.009092550
## [10,] 0.5788106 0.2647024 0.035307614
## [11,] 0.6917356 0.1216890 0.004567958
## [12,] 0.6160301 0.2556519 0.023976009
## [13,] 0.6353450 0.1691165 0.050358059
## [14,] 0.6585655 0.2276463 0.003412387
## [15,] 0.6228265 0.2665866 0.010810671
```

```
head(ACP_beans$ind$contrib[,1:3],15)
```

```
##             [,1]       [,2]         [,3]
##  [1,] 0.05441560 0.03199355 0.0091305972
##  [2,] 0.05124967 0.02868280 0.0021470156
##  [3,] 0.06069116 0.04643293 0.0106175378
##  [4,] 0.06113345 0.04631008 0.0102804225
##  [5,] 0.04905206 0.02724693 0.0005944211
##  [6,] 0.03803614 0.02345376 0.0709968404
##  [7,] 0.05142750 0.03784622 0.0188779519
##  [8,] 0.03801774 0.01958780 0.0172703477
##  [9,] 0.06667048 0.05854967 0.0061553982
## [10,] 0.05952184 0.05179923 0.0223784098
## [11,] 0.03142855 0.01052109 0.0012791669
## [12,] 0.05438781 0.04295108 0.0130466056
## [13,] 0.03627214 0.01837278 0.0177195818
## [14,] 0.05046982 0.03319853 0.0016118030
## [15,] 0.05582684 0.04547154 0.0059724093
```

```
head(ACP_beans$var$coord[,1:3],15)
```

```
##             [,1]       [,2]         [,3]
##  [1,] -0.8445434  0.4973990  0.133811960
##  [2,] -0.9284619  0.3616121 -0.010094165
##  [3,] -0.9719689  0.1980620  0.090147253
##  [4,] -0.7110414  0.6974623  0.002259431
##  [5,] -0.6758961 -0.6899601  0.143113777
##  [6,] -0.6802312 -0.6654714  0.177307079
##  [7,] -0.8466752  0.4950362  0.126583190
##  [8,] -0.8899813  0.4488583  0.056303341
##  [9,]  0.1629344  0.4426519  0.157635641
```

```
## [10,]  0.4087100  0.2010224  0.675069618
## [11,]  0.7316007  0.4353594  0.380930654
## [12,]  0.7018134  0.6855765 -0.147670411
## [13,]  0.6687730 -0.6723369  0.138046046
## [14,]  0.9352572  0.2747700 -0.063067994
## [15,]  0.7033075  0.6825773 -0.152597553
```

```
head(ACP_beans$var$cos2[,1:3],15)
```

```
##             [,1]       [,2]         [,3]
##  [1,] 0.71325347 0.24740576 1.790564e-02
##  [2,] 0.86204150 0.13076330 1.018922e-04
##  [3,] 0.94472363 0.03922856 8.126527e-03
##  [4,] 0.50557983 0.48645369 5.105027e-06
##  [5,] 0.45683548 0.47604491 2.048155e-02
##  [6,] 0.46271444 0.44285217 3.143780e-02
##  [7,] 0.71685887 0.24506084 1.602330e-02
##  [8,] 0.79206677 0.20147378 3.170066e-03
##  [9,] 0.02654763 0.19594072 2.484900e-02
## [10,] 0.16704389 0.04041001 4.557190e-01
## [11,] 0.53523954 0.18953784 1.451082e-01
## [12,] 0.49254206 0.47001519 2.180655e-02
## [13,] 0.44725737 0.45203692 1.905671e-02
## [14,] 0.87470609 0.07549858 3.977572e-03
## [15,] 0.49464142 0.46591171 2.328601e-02
```

```
head(ACP_beans$var$contrib[,1:3],15)
```

```
##            [,1]      [,2]         [,3]
##  [1,] 7.2094006 4.7587245 1.115491e+00
##  [2,] 8.7133154 2.5151657 6.347709e-03
##  [3,] 9.5490472 0.7545416 5.062689e-01
##  [4,] 5.1102835 9.3566903 3.180345e-04
##  [5,] 4.6175870 9.1564827 1.275966e+00
##  [6,] 4.6770101 8.5180371 1.958522e+00
##  [7,] 7.2458431 4.7136211 9.982246e-01
##  [8,] 8.0060271 3.8752461 1.974897e-01
##  [9,] 0.2683373 3.7688205 1.548050e+00
## [10,] 1.6884408 0.7772661 2.839052e+01
## [11,] 5.4100770 3.6456642 9.039992e+00
## [12,] 4.9785008 9.0405039 1.358511e+00
## [13,] 4.5207737 8.6947011 1.187201e+00
## [14,] 8.8413261 1.4521769 2.477960e-01
## [15,] 4.9997207 8.9615756 1.450679e+00
```

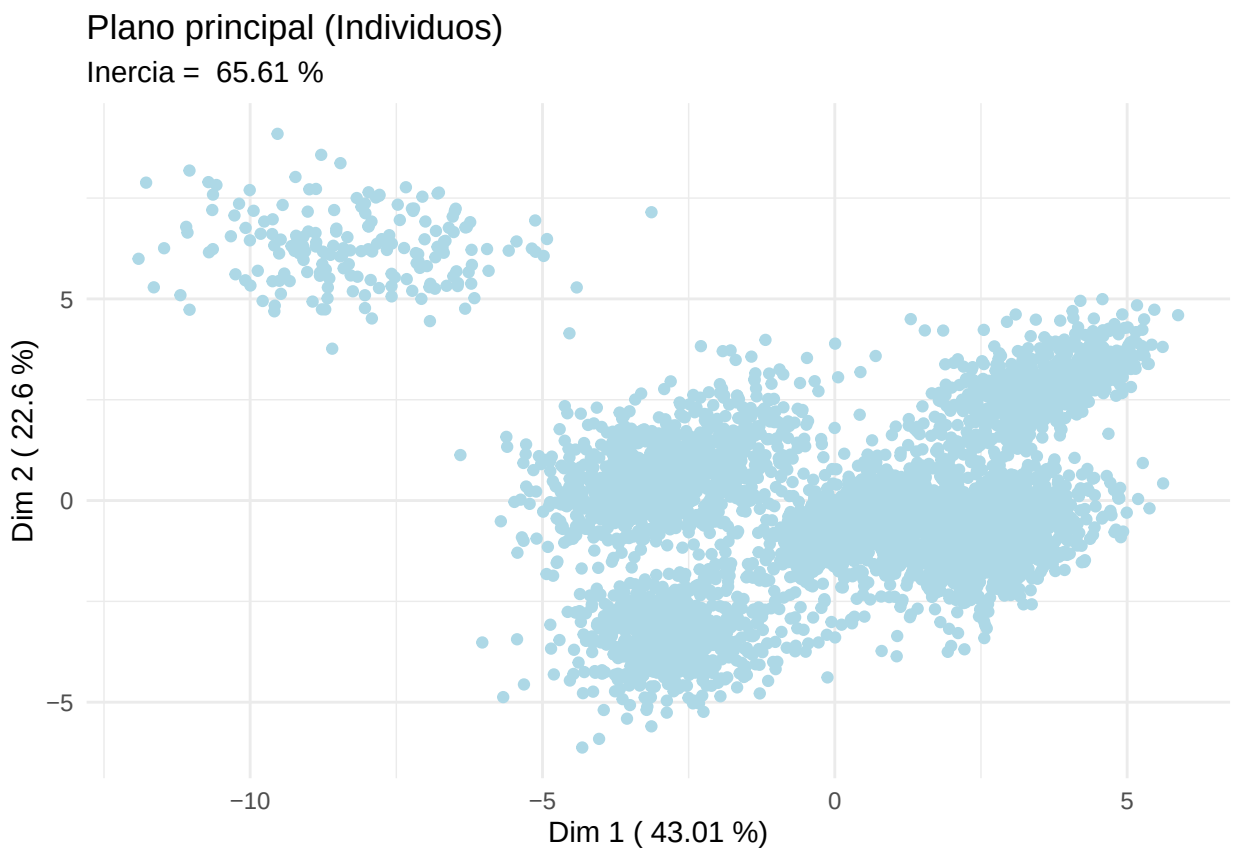
```
head(ACP_beans$inercias)
```

```
## [1] 43.014700 22.604318  6.979044  5.904004  5.658841  5.421096
```

Inciso 3

Plano principal

```
plano_principal(beans_datos)
```



Círculo de correlaciones

```
circulo_correlaciones(beans_datos)$variables
```

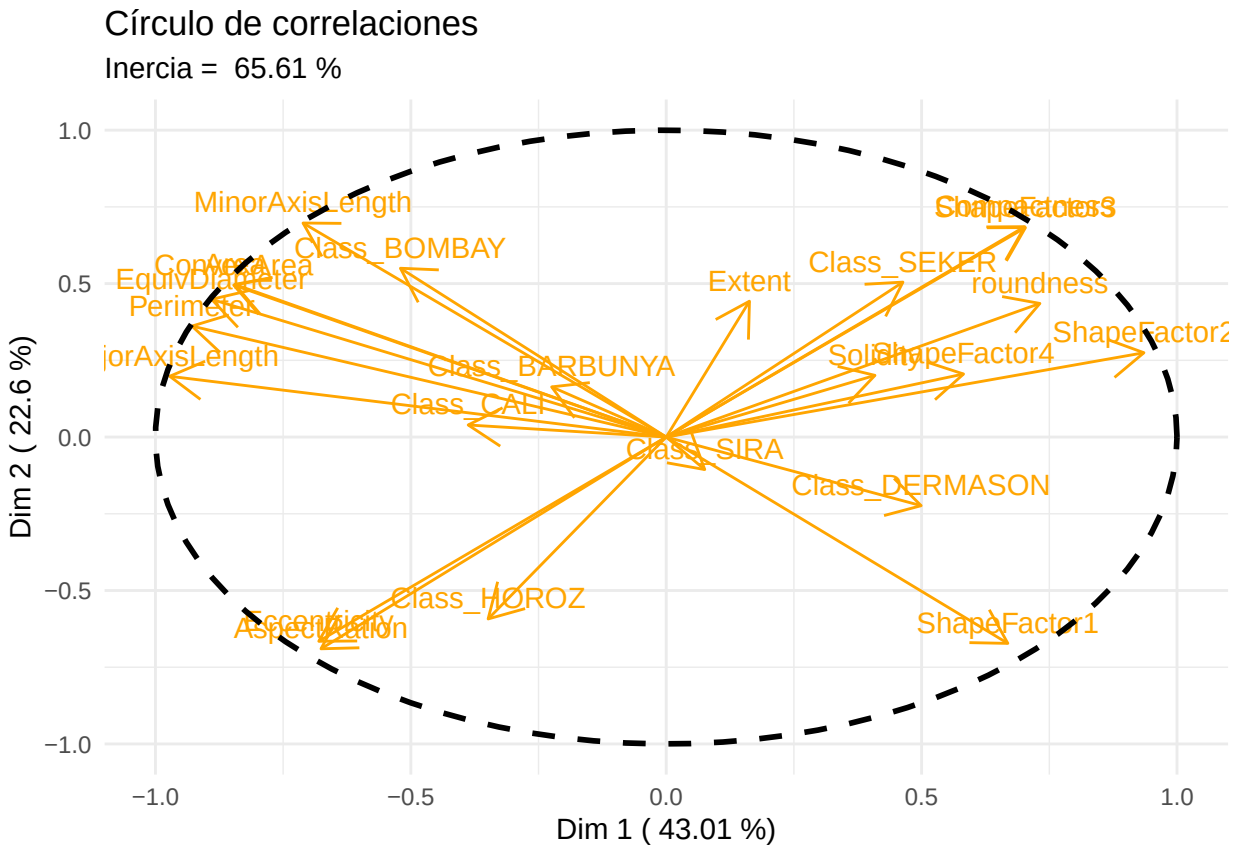
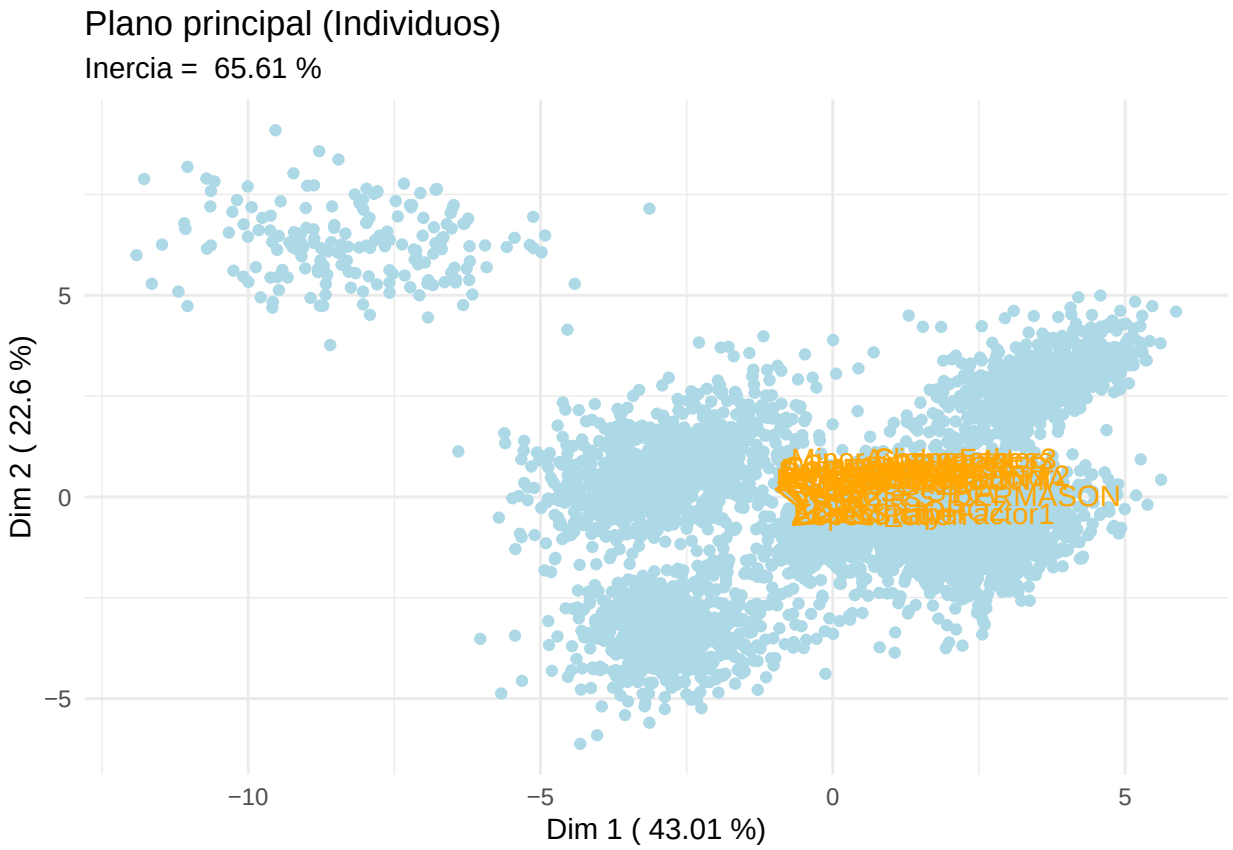


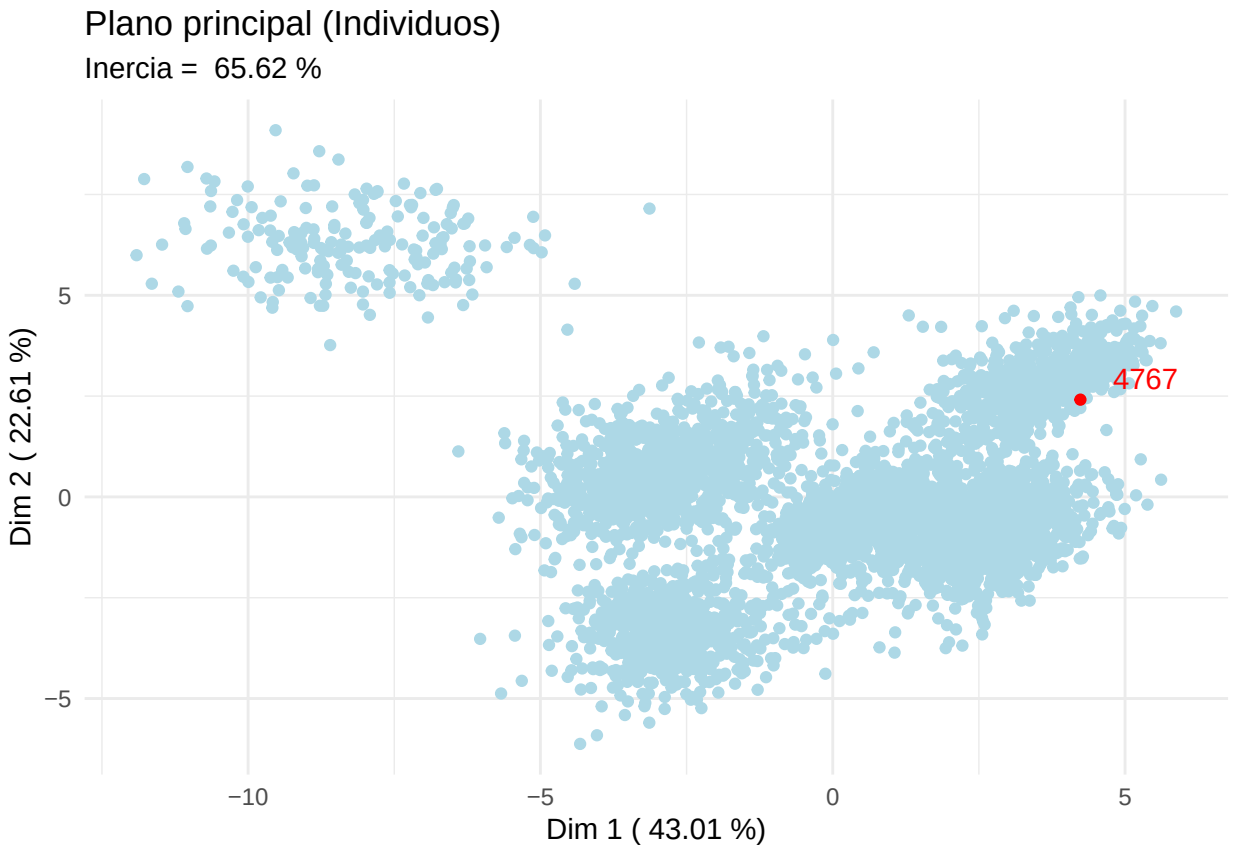
Gráfico Dual

```
grafico_dual(plano_principal(beans_datos),
             circulo_correlaciones(beans_datos), beans_datos)
```



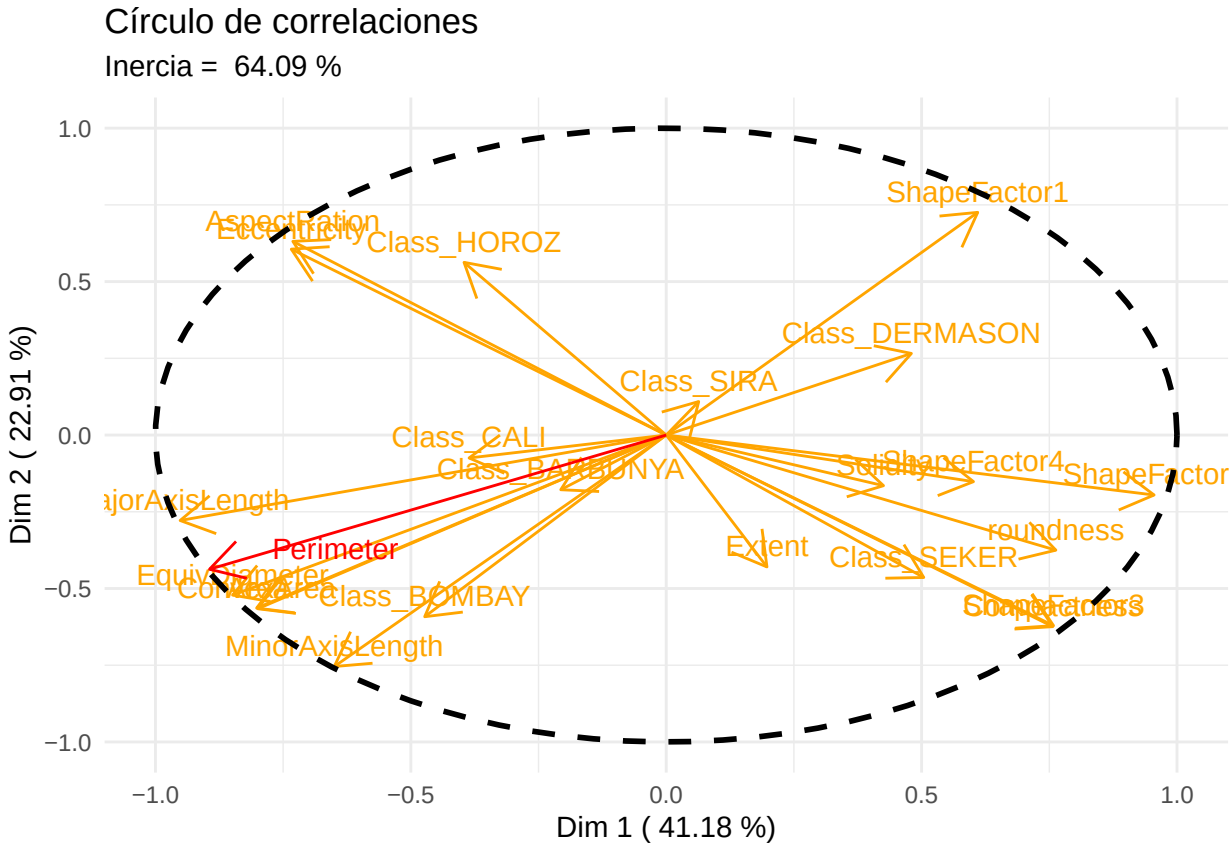
Inciso 4

```
fila_sup_beans <- beans_datos[6,]
coordenadasind_sup_beans <- ind_sup_proyeccion(fila_sup_beans, beans_datos[-6,])
coordenadasind_sup_beans
```



Inciso 5

```
columna_sup_beans <- as.matrix(beans_datos[,2])
colnames(columna_sup_beans) <- (colnames(beans_datos))[2]

coordenadasvar_sup_beans <- var_sup_proyeccion(columna_sup_beans,
                                               beans_datos[,-2])
coordenadasvar_sup_beans
```



Inciso 6

Dado que la base de datos `beans_datos` posee un total de 4767 individuos, al calcular la matriz H, se obtienen unas dimensiones de 4767x4767 lo cual genera que al pasarlo por la función `eigen()`, se tarde un periodo de tiempo elevado al calcular los vectores y valores propios. Por lo tanto, se decide no evaluar el código. Esto se corrobora en el inciso 7 donde la función `ACP()`, ejecuta a la función basada en la matriz de correlaciones R.

```
ACP_beans_H <- ACP_basado_en_H(centrar_y_reducir(beans_datos))
ACP_beans_H
```

Inciso 7

Este inciso no se imprime debido a que, por ser la matriz R más optima que la H, entonces se correría la función `ACP_funcion()`, que ya se ejecutó anteriormente.

```
ACP_beans_optimo <- ACP(beans_datos)
```