PEC1 Análisis de datos ómicos

Estudiante: Irene Valdivia Callejón

Tabla de contenidos

1. Abstract

2. Objetivos del estudio

3. Materiales y métodos

3.1 Elección del dataset y descarga

3.2 Creación del SummarizedExperiment

3.3 Exploración de los datos

3.4 Características generales y representaciones univariantes

3.5 Representaciones multivariante

4. Resultados

4.1 Elección del dataset, descarga y SummarizedExperiment

4.2 Exploración de los datos

5. Discusión y limitaciones y conclusiones del estudio

6. Repositorio

1. Abtract

2. Objetivos del estudio

El propósito de este estudio es seleccionar uno de los conjuntos de datos de metabolómica disponibles en el repositorio https://github.com/nutrimetabolomics/metaboData/ y realizar una exploración bioinformática de estos datos en R. Esta exploración incluirá el análisis del tipo de variables contenidas en los datos para entender la naturaleza del estudio, los paramétros de salida o "readouts" , resultados, y el número de sujetos

involucrados, entre otros aspectos. Además, se explorarán distintas formas de representar y visualizar los datos de metabolómica para identificar patrones, valores atípicos y comprender qué variables explican la mayor parte de la variabilidad en el conjunto de datos.

3.Materiales y Métodos

3.1 Elección del dataset y descarga

Para seleccionar el dataset desde el repositorio de github https://github.com/nutrimetabolomics/metaboData/ , el repositorio ha sido clonado en R siguiendo los siguientes pasos: File -> New Project -> Version control -> Git y luego he copiado la dirección al repositorio. En datasets, el conjunto de datos de la carpeta metabotypingPaper ha sido elegido como datos con los que trabajar en esta actividad. Los datos han sido importados a R mediante la función read.csv().

3.2 Creación del SummarizedExperiment

Para la creación del SummarizedExperiment se ha usado el paquete SummarizedExperiment perteneciente a Bioconductor. La matriz con todos los datos se han dado como input en la variable "assays" mientras que hemos extraído el nombre de las columnas y filas para darlas como input en las variables "ColData" y "RowData". Para el metadata hemos usado el archivo metadata que hemos encontrado en la carpeta juntos a los datos originales en el repositorio del github.

3.3 Exploracion de los datos

3.4 Características generales y representaciones univariantes

Para obtener información sobre características generales de nuestros datos, las funciones "colname()" y "rowname()" han sido usadas para visualizar cuales son los diferentes parámetros representados en las filas y las columnas de nuestros datos. Con la función "summary()" se ha obtenido más información sobre los datos que tenemos en cada columna, como por ejemplo el rango, la media, el mínimo o el máximo.

Con la función "hist()" se ha visualizado la distribución de las distintas variables numéricas. Para variables no numéricas como el género o el tipo de cirugía, se ha utilizado la función "table()" para obtener las frecuencias y posteriormente se han representado usando barplots con la funcion "barplot()".

3.5 Representaciones multivariante

Para las representaciones univariantes se ha seleccionado primero aquellas variables que son numérica, que no continene missing values (NA) y que no contengan solo ceros. Dejar estas variables en nuestros datos causaba problemas para realizar un principal component analysis (PCA).

3.5.1 PCA

Para realizar el PCA se ha usado la función prcomp(), con la opción de scale= TRUE. Para representar el gráfico de PCA se ha usado la función autplot() del paquete fortify. Algunas variables como género, tipo de cirugía o edad se han representado en el gráfico de PCA añdiendo estas en el argumento "colour".

Para representar que parámetros contribuyen a los distintos componentes y que variables contribuyen más al primer componente se ha usado el paquete factoextra, concretamente las funciones get_pca_var() y fviz_contrib().

3.5.2 Heatmap

Para represnetar los datos en un heatmap se ha usado la función heatmap() dentro del paquete Complex-heatmap que pertenece a bioconductor.

3.6 Creación del repositorio

Para crear un repositorio en github, el primer paso ha sido iniciar sesión con mi cuenta. En la esquina izquierda, se ha clicado en el botón de crear un nuevo repositorio. Como nombre del respositorio se ha elegido "MetaboData_PEC1". Posteriormente he subido los archivos indicados en las instrucciones de la PEC1 y añadido un README.md file.

4. Resultados

4.1 Elección del dataset, descarga y SummarizedExperiment

He clonado el repositorio de GitHub en R y he seleccionado el dataset de la carpeta 2018-MetabotypingPaper. El dataset, en formato .csv, ha sido leído en R y guardado en la variable metabo_data. Además, he encontrado otro archivo con el metadata de mi dataset, el cual también ha sido leído en R:

```
#Read data

metabo_data <- read.csv(
"C:/Users/irene/OneDrive/Documentos/metaboData/Datasets/2018-MetabotypingPaper/DataValues_S013.csv")

metadata_metabo <- read.csv(
  "C:/Users/irene/OneDrive/Documentos/metaboData/Datasets/2018-MetabotypingPaper/DataInfo_S013.csv")
```

He creado un "SummarizedExperiment" aportando como input la matrix con todos los datos. He extraído los nombres de las columnas y filas para introducirlo en los parámetros "colData" y "rowData". Finalmente, he guardado el archivo para subirlo al respositorio.

```
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")

BiocManager::install("SummarizedExperiment")

library(SummarizedExperiment)


col= colnames(metabo_data)
row= rownames(metabo_data)

se <- SummarizedExperiment(assays = list(counts = metabo_data),
                           colData = col,
                           rowData = row,
                           metadata = metadata_metabo)



save(se, file = "SummarizedExperiment.rda")
```

4.2 Exploración de los datos

El primer paso es obtener información sobre las dimensiones de nuestro dataset, de las columnas y filas que tenemos:

```
dim(se)
```

```
## [1]  39 696
```

```
head(colnames(se))
```

```
## [1] "X.1"      "SUBJECTS" "SURGERY"  "AGE"      "GENDER"   "Group"
```

```
head(rownames(se))
```

```
## [1] "1" "2" "3" "4" "5" "6"
```

Observamos que hay 39 filas y 696 columnas.Uso el comando head() para evitar ocupar el informe con los datos, ya que hay muchas columnas y ocuparía mucho espacio. Sin embargo, se podrían visualizar los nombres de todas las filas y columnas. Cuando exploramos los nombres de las columnas y las filas vemos que en las columnas tenemos multitud de parámetros, entre ellos, la columna de "Subjects" que hará referencia a los sujetos del estudio. Sin embargo, la mayoría de la información en las columnas hace referencia a parámetros metabólicos. Por otro lado, con el nombre de las filas vemos que solo hay números, indicando que probablemente hay 39 sujetos y que en las diferentes columnas habrá información sobre estos sujetos, así como distintas mediciones.

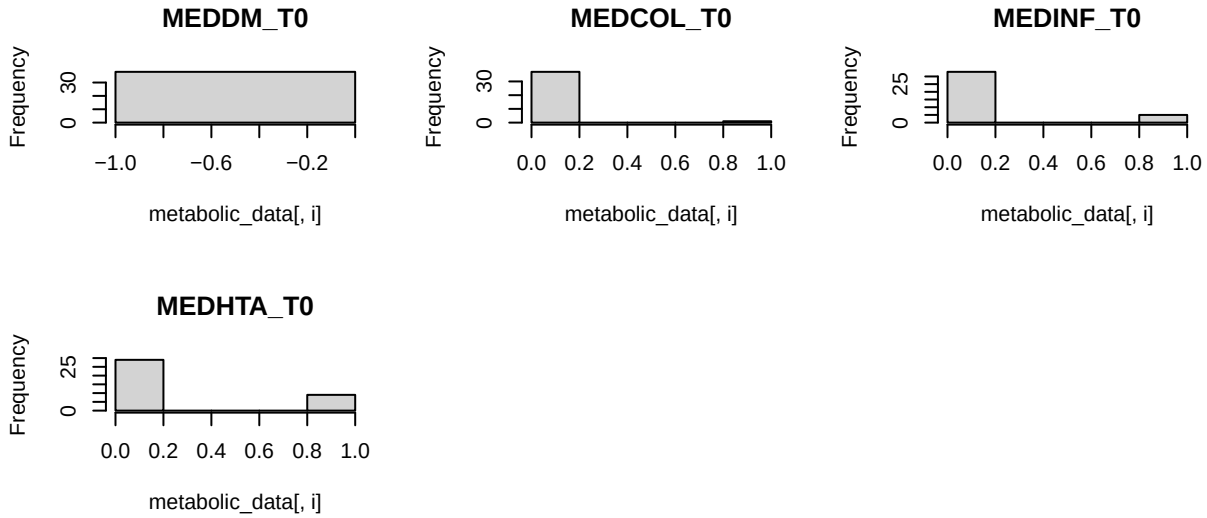
```
metabolic_data <- assay(se, "counts")


summary_result <- summary(metabolic_data [1:6])
head(summary_result)
```

```
##       X.1          SUBJECTS      SURGERY              AGE
##  Min.   : 1.0   Min.   : 1.0   Length:39          Min.   :19.00
##  1st Qu.:10.5   1st Qu.:10.5   Class :character   1st Qu.:35.00
##  Median :20.0   Median :20.0   Mode  :character   Median :41.00
##  Mean   :20.0   Mean   :20.0                      Mean   :40.79
##  3rd Qu.:29.5   3rd Qu.:29.5                      3rd Qu.:46.00
##  Max.   :39.0   Max.   :39.0                      Max.   :59.00
##     GENDER              Group
##  Length:39          Min.   :1.000
##  Class :character   1st Qu.:1.000
##  Mode  :character   Median :1.000
##                     Mean   :1.385
##                     3rd Qu.:2.000
##                     Max.   :2.000
```

Usando la función "summary" podemos obtener información sobre los valores que tenemos para cada parámetro como por ejemplo el mínimo, máximo, media, etc. He seleccionado las primeras columnas para no ocupar mucho espacio en el informe.

Con la función hist() podemos crear histogramas y observar las distribuciones de distintas variables individualmente.

```r
opt <- par(mfrow = c(3, 3))


for (i in 7:10) {
  hist(metabolic_data[, i], main = names(metabolic_data)[i])
}

par(opt)
```



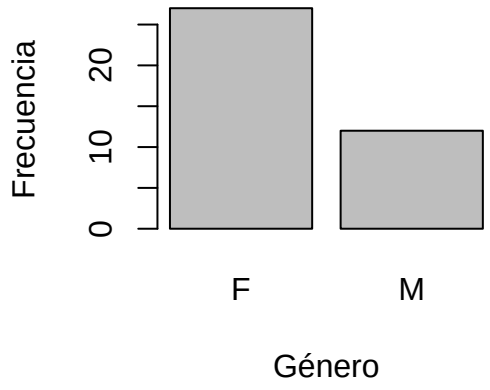
Anteriormente hemos visto que además de las mediciones metabólicas también hay información acerca de los sujetos, como por ejemplo, la edad, el tipo de cirugía o el género.Esta información puede ser importante para interpretar nuestros datos. Por ejemplo, cuando hacemos un principal component analysis o PCA podría ser que encontráramos patrones influenciados por alguna de estas variables. Quizá la pregunta de investigación incluye saber si hay diferencias metabólicas entre géneros, edad o entre pacientes que se han sometido a un tipo de cirugía u otro. De ser así, podría ser importante conocer la distribución de edad de tus sujetos, el género u otra información clínica.

Por ello, vamos a explorar estas varaibles:

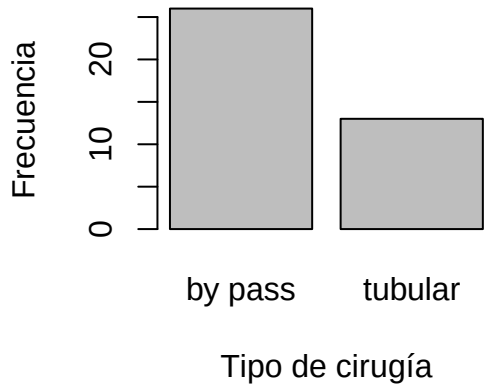
```r
#Género
freq_gender <- table(metabolic_data$GENDER)
freq_gender
```

```
##
##  F  M
## 27 12
```

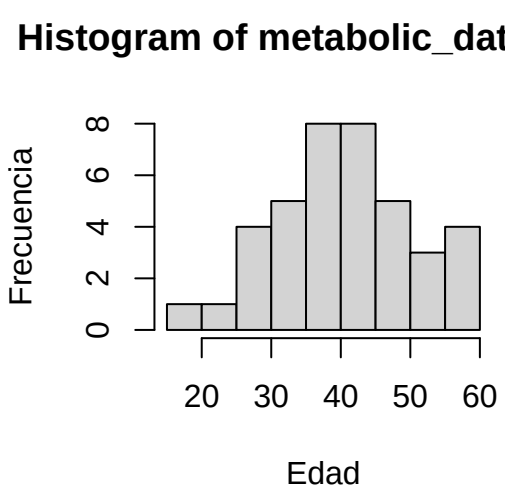
```r
barplot(freq_gender, xlab= "Género", ylab= "Frecuencia")
```




```r
#Cirugía

freq_ciru <- table(metabolic_data$SURGERY)

barplot(freq_ciru, xlab= "Tipo de cirugía", ylab= "Frecuencia")
```




```r
#Edad

hist(metabolic_data$AGE, xlab= "Edad", ylab= "Frecuencia")
```




```
max(metabolic_data$AGE)
```

```
## [1] 59
```

```
min(metabolic_data$AGE)
```

```
## [1] 19
```

Vemos que entre los sujetos de nuestro dataset hay 59 mujeres mientras que solo 19 son hombres. El rango de edad va de 19 a 59 años, sin embargo, la mayoría de los pacientes están sobre los 30 años de edad. También podemos ver que hay un mayor número de sujetos que se han sometido a una cirugía de tipo by pass en comparación a la tubular.

Ya que tenemos un gran número de parámetros, utilizamos herramientas como PCA para reducir la dimensionalidad de nuestros datos y poder representarlos reteniendo la mayor parte de la variabilidad. Ya que muchas variables contienen solo 0 or NAs, o bien no son numéricas, voy primero a quitar estas columnas para poder realizar mi PCA.

```
#Quito las primeras columnas ya que no son sobre datos metabólicos
subset_data <- metabolic_data[, 10:696]

#Quito columnas con NAs, no numéricas y con solo ceros


#Quitar columnas no numéricas
numeric_data <- subset_data[, sapply(subset_data, is.numeric)]

#Quitar columnas con solo ceros
numeric_data <- numeric_data[, colSums(numeric_data != 0, na.rm = TRUE) > 0]

#Quitar columnas con NAs
numeric_data <- numeric_data[ , colSums(is.na(numeric_data))==0]
```

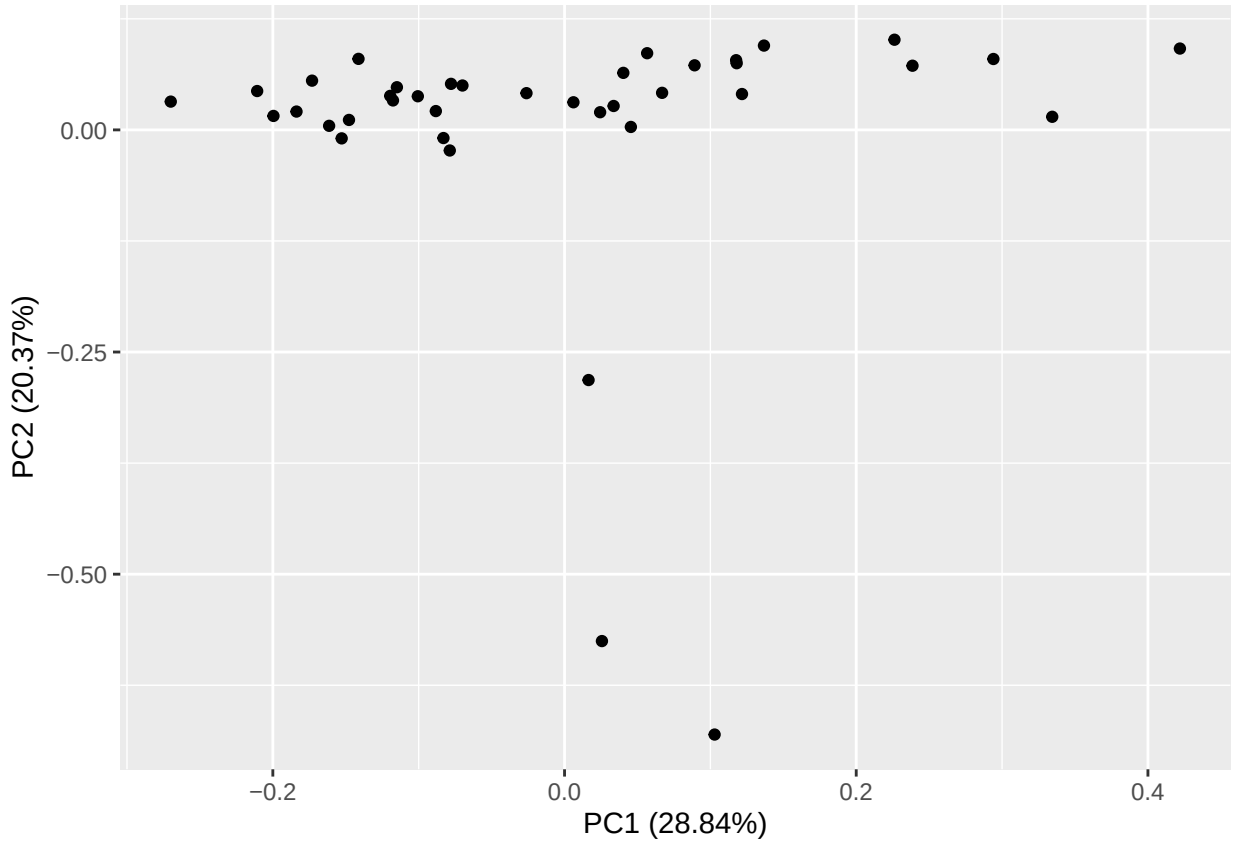
```
library(ggfortify)
```

```
## Warning: package 'ggfortify' was built under R version 4.3.3
```

```
## Loading required package: ggplot2
```

```
## Warning: package 'ggplot2' was built under R version 4.3.3
```

```
pca_result <- prcomp(numeric_data, scale. = TRUE)

#Representamos PCA
autoplot(pca_result)
```

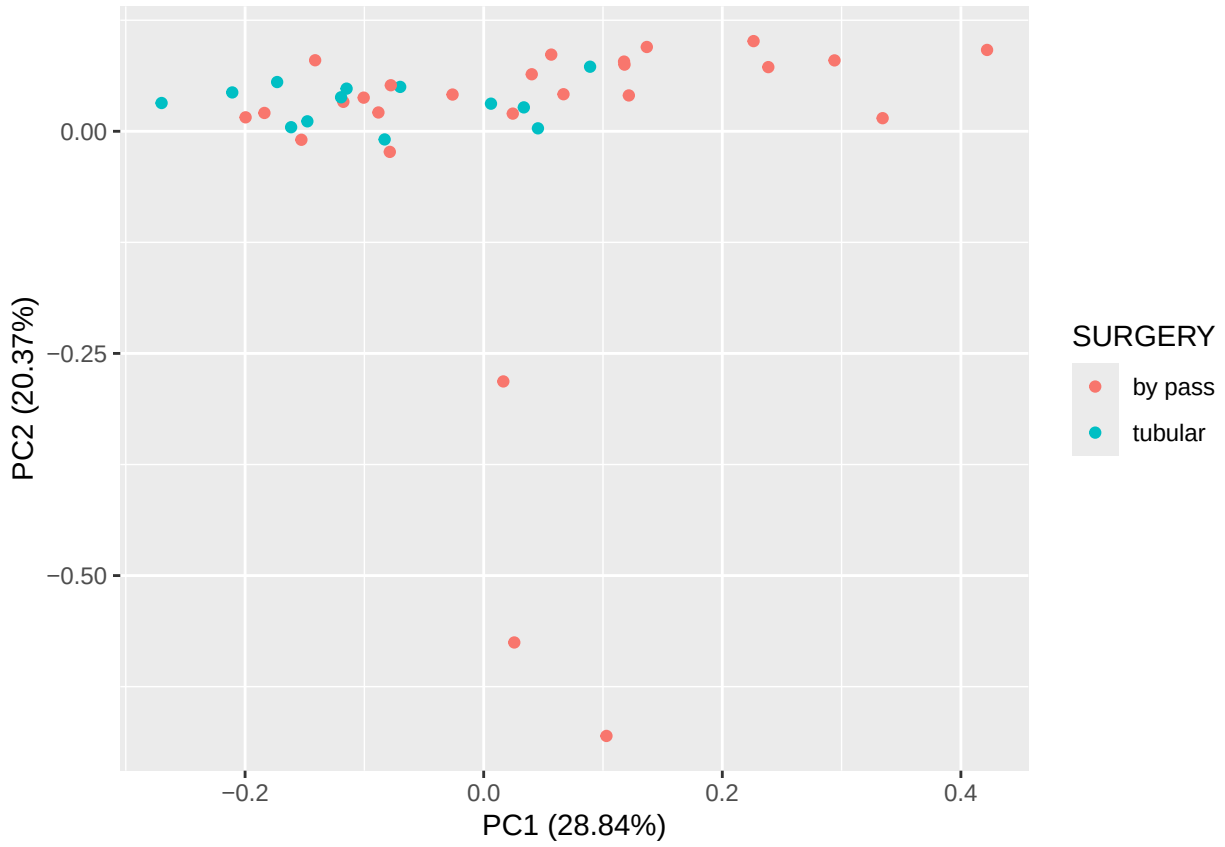


Obsevamos que en nuestra representación de los dos primeros componentes principales, que explicarían la mayor parte de la variabilidad de los datos, no se observan grupos que están claramente separados. Una utilidad de este tipo de gráficos es para visualizar si hay batch effects en el caso de que veamos que hay alguna agrupación en base a un factor técnico, por ejemplo, en base al día que se procesaron o el equipamiento del laboratorio usado.

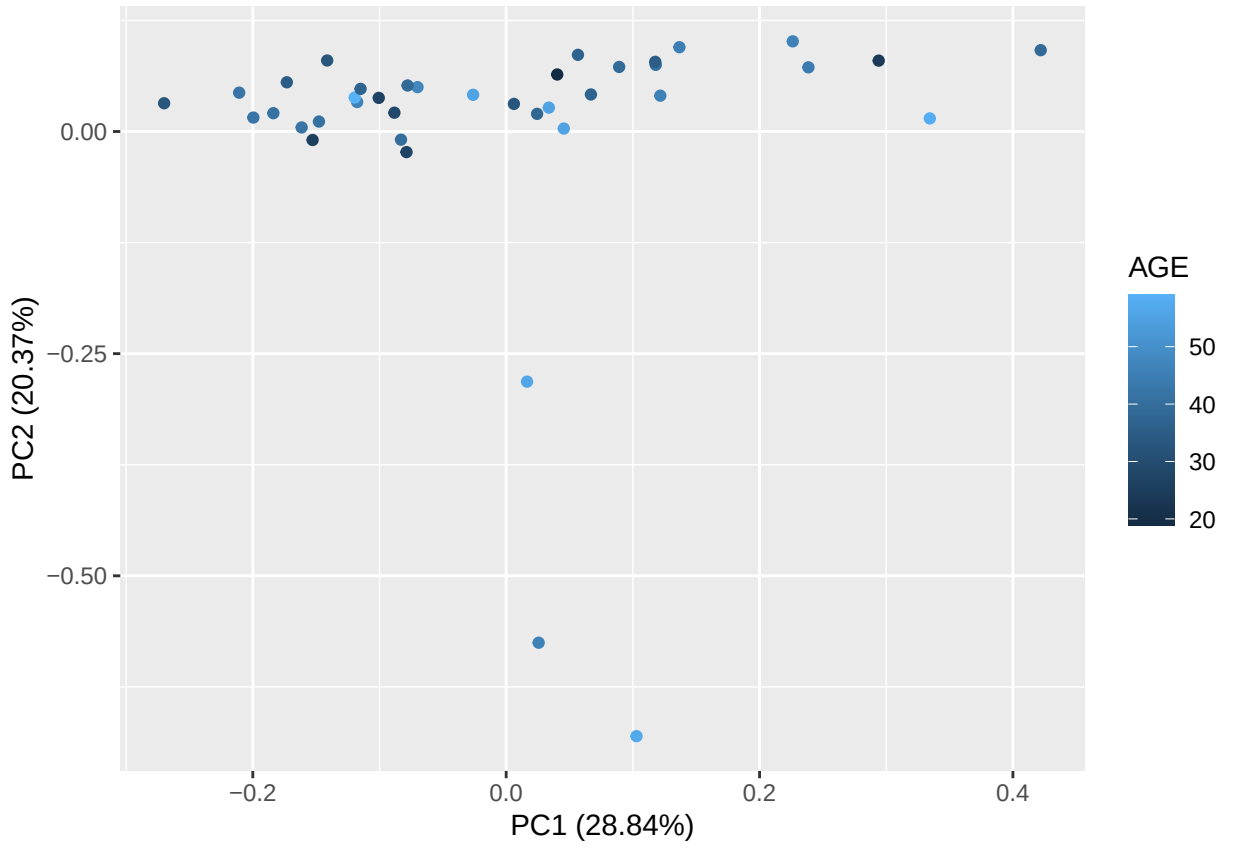
El hecho de que no haya grupos claramente separados podría indicar que no podríamos diferenciar entre género, edad o tipo de cirugía de los sujetos en base a su perfil metabólico. Sin embargo, si que vemos que hay tres pacientes que se encuentran más alejados que del resto, indicando que el perfil metabólico de estos sujetos es bastante diferente al resto.

Vamos a explorar más los datos:

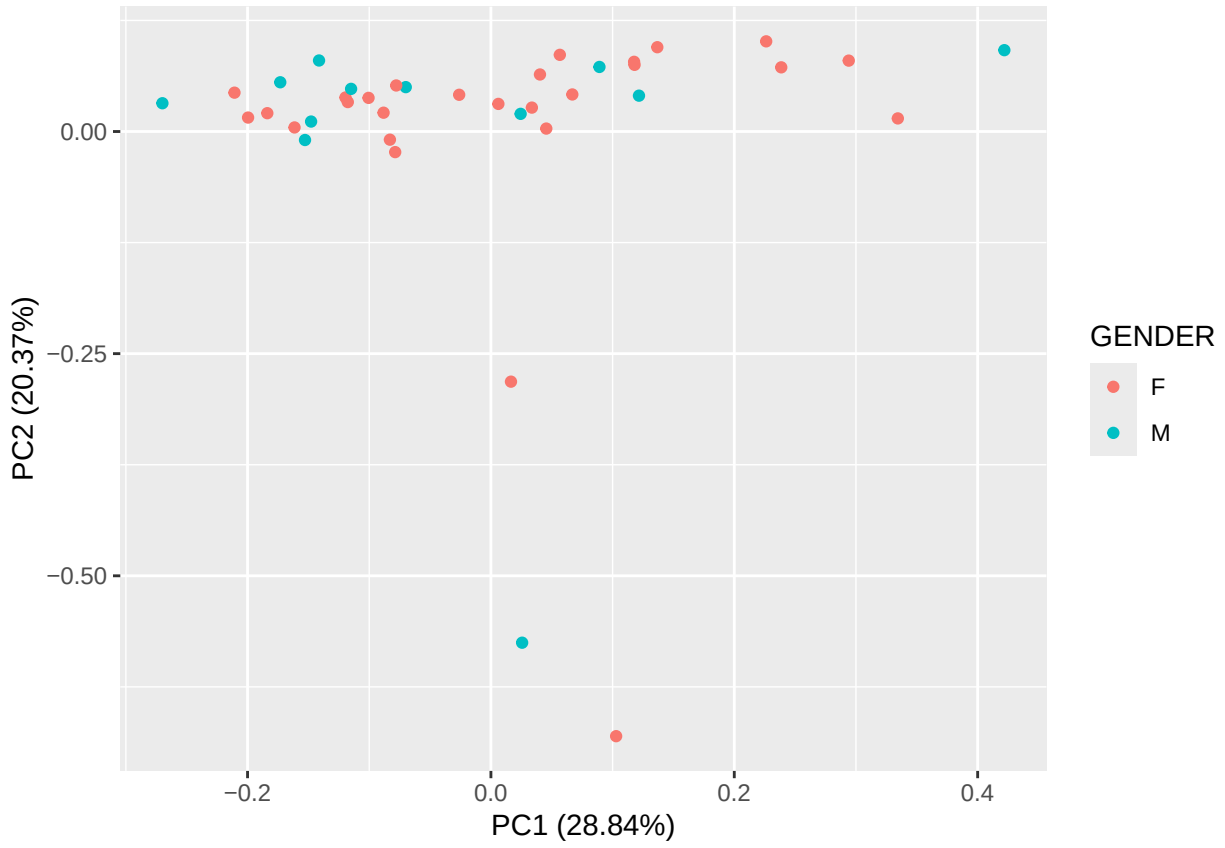
```
autoplot(pca_result, data = metabolic_data , colour = "SURGERY")
```




```
autoplot(pca_result, data = metabolic_data , colour = "AGE")
```



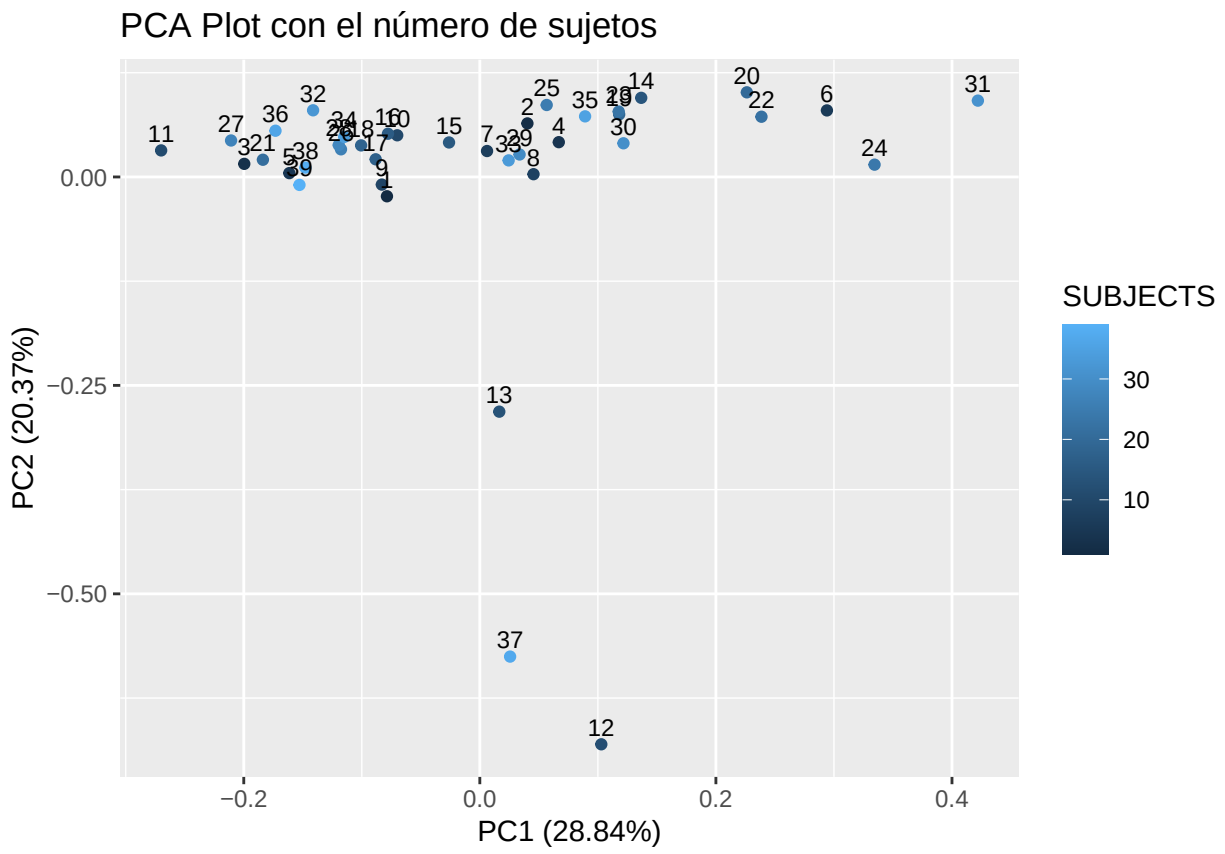

```
autoplot(pca_result, data = metabolic_data , colour = "GENDER")
```



En las representaciones anteriores vemos que no hay una clara agrupación en base a los parámetros de cirugía, edad o género. Si que vemos que los tres sujetos más alejados del resto todos pertenencen al grupo que ha tenido un bypass.

Podríamos ver qué sujetos son los que no se agrupan con el resto para explorarlos más a fondo y decidir si estas diferencias son relevantes para nuestra cuestión biológica o si determinamos que representan outliers y queremos eliminarlos de nuestros datos en el caso de que estuvieran afectando a la representación de nuestros datos.

```
library(ggplot2)
library(ggfortify)

pca_plot <- autoplot(pca_result, data = metabolic_data, colour = "SUBJECTS") +
  geom_text(aes(label = SUBJECTS), size = 3, vjust = -0.5) +
  labs(title = "PCA Plot con el número de sujetos")

pca_plot
```



Otra información que puede resultar importante es ver qué parámetro contribuyen a los distintos "principal components", en especial, ver qué parámetros contribuyen más al primer componente principal.

```
library(factoextra)
```

```
## Warning: package 'factoextra' was built under R version 4.3.3
```

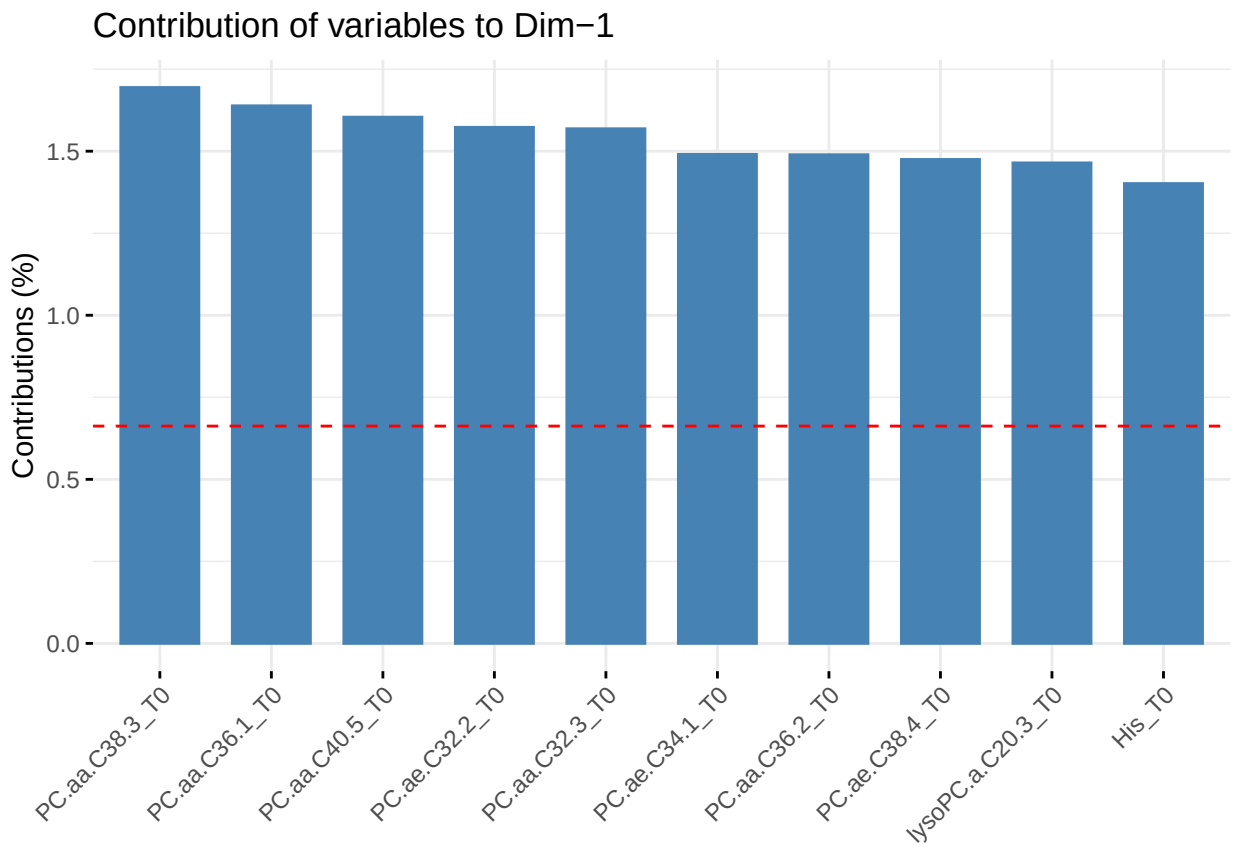
```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
var <- get_pca_var(pca_result)

head(var$contrib, 4)
```

```
##               Dim.1        Dim.2    Dim.3     Dim.4    Dim.5      Dim.6
## GLU_T0  0.043657049 0.0002430149 1.036285 2.3079167 1.369989 5.60866140
## INS_T0  0.002040775 0.2007503772 1.030775 0.5470126 3.265502 0.00696922
## HOMA_T0 0.004074473 0.1086972959 1.396880 1.4773866 3.416452 1.26118678
## PESO_T0 0.120483335 0.0160373408 2.499744 0.1126856 2.093918 1.22610499
##                Dim.7       Dim.8     Dim.9      Dim.10    Dim.11    Dim.12
## GLU_T0  0.1031395037 0.540749794 0.6320186 0.01437712 2.0020754 0.4065576
## INS_T0  0.4031538974 0.290592300 2.3706629 1.79585454 1.8748651 4.0591169
## HOMA_T0 0.0002909695 0.001915203 2.3331063 0.54671457 3.1296189 1.8383590
## PESO_T0 0.0029828066 1.293201515 6.1118513 0.01110599 0.2215412 0.1049815
##            Dim.13       Dim.14     Dim.15       Dim.16     Dim.17       Dim.18
## GLU_T0  0.2071960 0.0058082222 0.03618735 0.140228490 0.02692011 0.009172711
```

```
## INS_T0  0.5021080 0.0863685332 4.05184409 0.309351835 0.01020789 3.657488817
## HOMA_T0 0.5425595 0.0002163081 2.62772685 0.041897079 0.13891137 2.595252540
## PESO_T0 1.0495714 0.1334291641 0.04096700 0.002951068 2.57035436 0.362296106
##              Dim.19    Dim.20       Dim.21    Dim.22     Dim.23     Dim.24
## GLU_T0  2.675566278 0.2575300 0.4852713939 2.2660921 0.48342171 1.22132896
## INS_T0  0.845081034 0.2180083 0.0005247508 0.2865654 0.12223806 0.06447722
## HOMA_T0 0.004828006 0.1158472 0.4706417298 0.2708969 0.25046007 0.03044844
## PESO_T0 0.235919280 3.7521787 0.0079269272 0.6802236 0.02448146 0.07506095
##              Dim.25    Dim.26       Dim.27    Dim.28     Dim.29    Dim.30
## GLU_T0  0.004786221 1.8163262 0.4666529937 0.4850577 0.33399184 0.8403920
## INS_T0  0.927415699 2.2858316 1.4618519388 1.4840930 0.50785741 0.9273149
## HOMA_T0 0.624385442 0.1849396 1.4661288713 0.4042200 0.05540465 0.6386103
## PESO_T0 0.066364242 0.2519097 0.0002882577 1.0055359 0.07473219 0.8082849
##            Dim.31     Dim.32       Dim.33      Dim.34     Dim.35      Dim.36
## GLU_T0  2.5086429 1.44148117 0.3533509049 1.017195638 1.60648437 1.049369850
## INS_T0  0.2643271 0.02033572 0.0215726634 0.004008379 0.01743042 0.003231946
## HOMA_T0 0.6485547 0.01889475 0.0293812940 0.039262639 0.02566397 0.193112973
## PESO_T0 0.1615692 3.04964811 0.0001691348 0.022199893 1.54683490 0.190634288
##             Dim.37    Dim.38    Dim.39
## GLU_T0  0.30670919 2.1999246 6.6106447
## INS_T0  0.09559928 0.7168105 0.5117516
## HOMA_T0 0.01481303 0.2032558 5.8778037
## PESO_T0 1.30661364 0.1161929 0.5605725
```

```
#Contribución se las variables a PC1
fviz_contrib(pca_result, choice = "var", axes = 1, top = 10)
```

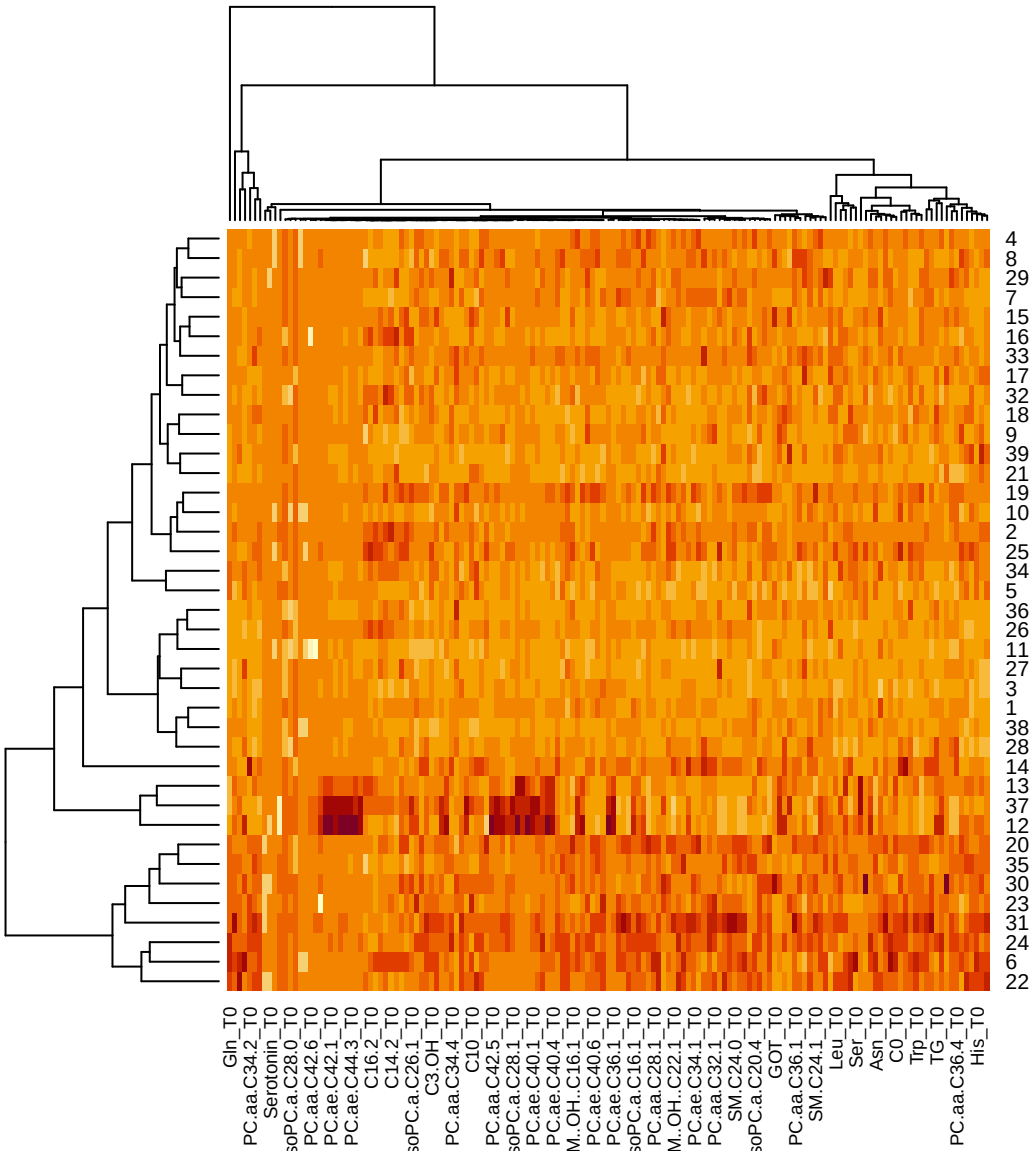


Con este código vemos que parámetros metabólicos son los que pueden explicar la mayor parte de la variabilidad de nuestros datos.

Otra forma de explorar nuestros datos es mediante un heatmap y dendograma. En los heatmaps se representan la mayor o menor expresión de un gen o bien la presencia de un metabolito en este caso, usando un rango de colores. Con el dendograma podemos encontrar que parámetros muestran similitudes.

```
if (!requireNamespace("BiocManager", quietly = TRUE))
    install.packages("BiocManager")
BiocManager::install("ComplexHeatmap")
```

```
matrix <- as.matrix(numeric_data)
heatmap(matrix, scale = "column")
```



Al mostrar el heatmap de nuestros datos, vemos que los colores más oscuros indicarían una mayor presencia de metabolitos. Además, el dendrograma muestra que ciertos parámetros están más relacionados entre sí, al igual que algunos pacientes. De manera interesante, observamos que los sujetos 37, 13 y 12 se agrupan en el

dendrograma, similar al gráfico de PCA, donde estos pacientes también aparecían cercanos entre sí y alejados del resto. Además vemos que estos tienen una expresión muy alta de ciertos parámetros, especialmente 12 y 27, en comparación con el resto de pacientes, lo cual también podría ser la razón por la cual se encuentran tan alejados en el gráfico de PCA.

5. Discusión y limitaciones y conclusiones del estudio

En este estudio hemos explorado datos metabólicos utilizando R. El primer paso fue revisar las variables presentes en el conjunto de datos, identificando que se trata de un estudio de 39 pacientes, de los cuales 27 son mujeres y 12 hombres, con un rango de edad de entre 19 y 59 años. Una de las columnas proporciona información clínica sobre los sujetos, indicando si han sido sometidos a una cirugía de tipo bypass o tubular. Las demás columnas contienen información sobre diversos parámetros metabólicos, que, según la descripción de los datos incluida en la carpeta, fueron registrados en cinco puntos temporales diferentes.

A partir de representaciones de parámetros metabólicos individualmente, quedó claro que, en datos ómicos, es necesario adoptar un enfoque distinto. Visualizar y analizar cada variable individualmente se convierte en un proceso tedioso que puede hacer perder la visión general del conjunto y la ventaja de trabajar con una gran cantidad de datos. Por esta razón, análisis como el de componentes principales (PCA) o los mapas de calor (heatmaps) son útiles para obtener una visión global de la estructura de los datos. En nuestro caso, el PCA no mostró agrupaciones claramente diferenciadas. Sin embargo, sería interesante investigar más a fondo si se observan agrupaciones de muestras en base a algún parámetro. La pregunta biológica podría ser sobre si hay o no una diferencia metabólica en pacientes sometidos a distintos tipos de cirugía. En este caso, no parece que hayamos encontrado una agrupación clara pero un estudio más riguroso sería necesario. En la descripción original del conjunto de datos, se menciona que los valores se recogieron en cinco momentos temporales distintos. Considerar estos puntos temporales podría ofrecer nuevos enfoques y, posiblemente, resultados diferentes al analizar las tendencias en el tiempo. Con el heatmap y dendograma, hemos podido visualizar la presencia de los distintos parámetros y hemos visto algunas agrupaciones que podrían ser interesantes de estudiar.

También es importante subrayar las limitaciones de este estudio. En primer lugar, los datos no son originales de nuestra investigación, por lo que carecemos de información que sería esencial para interpretar los resultados con precisión, como la hipótesis o cuestión biológica planteada en el estudio original. Además, no contamos con una descripción detallada de cada columna en los metadata, lo que dificulta identificar con exactitud los parámetros medidos. Algunas columnas contienen únicamente valores de 0 o 1, pero se desconoce qué representa cada número, mientras que otras contienen numerosos valores que faltan (NAs) sin aclarar el motivo de su ausencia.

En general, es fundamental destacar que, si se hacen datos disponibles públicamente, es indispensable proporcionar toda la información relevante y un enlace al estudio original. Esto facilita el trabajo de otros investigadores interesados en utilizar estos datos y aumenta la utilidad y reproducibilidad del estudio.

6. Repositorio

El link al repositorio de github con los documentos que se piden en el PEC1: https://github.com/ival9/MetaboData_PEC1.git