# Informe_PEC1

Javier Gancedo Verdejo

## Contents



## Abstract

En este estudio se analizan los efectos de la cirugía bariátrica sobre diversos metabolitos. Para ello, se empleó un dataset con datos metabólicos utilizado en un estudio de 2018. Dichos datos se extrajeron y prepararon para generar un objeto contenedor en formato SummarizedExperiment de Bioconductor. De esta forma el contenedor generado contenía la matriz de valores, así como los metadatos del dataset, de las filas (características de las variables) y de las columnas (información de las muestras).

Después de generar el contenedor, se llevó a cabo una exploración de los datos mediante un análisis univariante y multivariante. En el análisis univariante nos centramos en algunas variables consideradas relevantes para este tipo de estudios. Inicialmente nos centramos en el peso o el BMI, para los cuales los resúmenes estadísticos y los histogramas mostraban una reducción progresiva de los mismos tras la cirugía bariátrica. También nos centramos en los valores de los distintos tipos de colesterol (Colesterol total, LDL, HDL y VLDL), donde las cajas de bigotes mostraban que, tras la cirugía, se producía una reducción significativa de los niveles de todos ellos excepto el VLDL.

El análisis multivariante se llevó a cabo mediante el análisis de componentes principales (PCA) y el agrupamiento jerárquico. En los PCAs las dos primeras componentes principales explicaban más del 67% de la

variabilidad, y se identificaron que los niveles al inicio del estudio de algunos aminoácidos (como la glutamina y la alanina entre otros) tenían una gran contribución en la variabilidad. El agrupamiento jerárquico reveló dos grupos de muestras distintos, y con la información aportada por los PCA se observó que las muestras con niveles más altos de glutamina en el momento de la operación se agrupaban.

Además, el código de R y los datos utilizados en este estudio se almacenaron en un repositorio de GitHub, haciéndolos accesibles para garantizar la reproducibilidad de los análisis.

# Objetivos del estudio

- Analizar el impacto de la cirugía bariátrica en diferentes metabolitos y parámetros sanitarios.
- Adaptar datos metabólicos para generar un contenedor SummarizedExperiment de Bioconductor.
- Explorar las diferentes variables mediante un análisis univariante basado en resúmenes estadísticos y gráficos descriptivos.
- Aplicar un análisis multivariante mediante análisis de componentes principales (PCA), para identificar factores latentes que explican la variabilidad y posibles efectos batch.
- Explorar los posibles subgrupos dentro del conjunto de datos mediante agrupaciones jerárquicas.
- Facilitar la reproducibilidad de los análisis y resultados empleando el repositorio GitHub.

# Materiales y Métodos

**Obtención y preparación de los datos**: Se utilizaron los datos metabólicos del dataset "2018-MetabotypingPaper" obtenidos del repositorio de GitHub: https://github.com/nutrimetabolomics/metaboData/. Se importaron los archivos .csv DataInfo_S013 y DataValues_S013. De dichos objetos se extrajeron la matriz de valores (values), y los metadatos de las filas (features), columnas (samples) y del dataset (metadata), y se adaptaron para poder generar posteriormente un contenedor con ellos.

**Generación de un contenedor**: Se generó un contenedor con los datos y metadatos. Para ello se empleó la función SummarizedExperiment() del paquete SummarizedExperiment de Bioconductor.

**Análisis univariante**:

- **Análisis de parámetros**: Las variables de peso y BMI se analizaron el tiempo utilizando resúmenes estadísticos (función summary()) e histogramas (función hist()).
- **Análisis de metabolitos (Colesterol)**: Se estudiaron los niveles de Colesterol total, LDL, HDL y VLDL con el tiempo tras la cirugía con diagramas de cajas (función boxplot()).

**Análisis multivariante**:

- **Análisis de componentes principales (PCA)**: Para realizar el PCA se eliminaron las variables con valores faltantes (función na.omit()). No se llevaron a cabo correcciones previas al PCA. El PCA se realizó con la función prcomp(), y se vizualizó con la función plot(). A partir de los resultados se estudiaron las contribuciones de las variables a la variabilidad, así como el efecto batch.
- **Agrupación jerárquica**: La agrupación se llevó a cabo utilizando la función hclust() y los métodos "average" y "complete". Se visualizó con la función plot(), y se empleó para determinar subgrupos dentro del conjunto de muestras.

**Repositorio de datos**: Se utilizó el respositorio GitHub (github.com) para almacenar el código de R y los datos y metadatos utilizados en este estudio.

# Resultados

## Obtención y preparación de los datos

En primer lugar instalamos los paquetes que vamos a necesitar. En este caso, para cread un contenedor del tipo SummarizedExperiment instalamos Bioconductor y el paquete SummarizedExperiment.

A continuación clonamos el repositorio de github indicado en el enunciado, el cual contiene los datasets de metabolómica. En mi caso el dataset denominado "2018-MetabotypingPaper" ha sido el dataset seleccionado.

A continuación utilizamos el paquete readr para cargar los archivos de dicho dataset que se encuentran en formato .csv.

```
## # A tibble: 6 x 4
##   ...1     VarName  varTpe    Description
##   <chr>    <chr>    <chr>     <chr>
## 1 SUBJECTS SUBJECTS integer   dataDesc
## 2 SURGERY  SURGERY  character dataDesc
## 3 AGE      AGE      integer   dataDesc
## 4 GENDER   GENDER   character dataDesc
## 5 Group    Group    integer   dataDesc
## 6 MEDDM_T0 MEDDM_T0 integer   dataDesc
```

```
## # A tibble: 6 x 696
##    ...1 SUBJECTS SURGERY   AGE GENDER Group MEDDM_T0 MEDCOL_T0 MEDINF_T0
##   <dbl>    <dbl> <chr>   <dbl> <chr>  <dbl>    <dbl>     <dbl>     <dbl>
## 1     1        1 by pass    27 F          1        0         0         0
## 2     2        2 by pass    19 F          2        0         0         0
## 3     3        3 by pass    42 F          1        0         0         0
## 4     4        4 by pass    37 F          2        0         0         0
## 5     5        5 tubular    42 F          1        0         0         0
## 6     6        6 by pass    24 F          2        0         0         0
## # i 687 more variables: MEDHTA_T0 <dbl>, GLU_T0 <dbl>, INS_T0 <dbl>,
## #   HOMA_T0 <dbl>, HBA1C_T0 <dbl>, HBA1C.mmol.mol_T0 <dbl>, PESO_T0 <dbl>,
## #   bmi_T0 <dbl>, CC_T0 <dbl>, CINT_T0 <dbl>, CAD_T0 <dbl>, TAD_T0 <dbl>,
## #   TAS_T0 <dbl>, TG_T0 <dbl>, COL_T0 <dbl>, LDL_T0 <dbl>, HDL_T0 <dbl>,
## #   VLDL_T0 <dbl>, PCR_T0 <dbl>, LEP_T0 <dbl>, ADIPO_T0 <dbl>, GOT_T0 <dbl>,
## #   GPT_T0 <dbl>, GGT_T0 <dbl>, URICO_T0 <dbl>, CREAT_T0 <dbl>, UREA_T0 <dbl>,
## #   HIERRO_T0 <dbl>, TRANSF_T0 <dbl>, FERR_T0 <dbl>, Ile_T0 <dbl>, ...
```

Los datos cargados, dado que no vienen en un formato adecuado para crear un contenedor del tipo SummarizedExperiment, se han tenido que adaptar y extraer de ellos la información necesaria. Del Dataframe DataInfo_S013 se puede extraer el metadata de las filas (features), mientras que del dataframe DataValues_S013 se puede extraer la matrix de valores y el metadata de las columnas (samples).

Además, podemos cargar la información que aparece en la descripción del dataset como metadata.

## Creación del Contenedor "Summarized Experiment"

Una vez que tenemos todos los datos y metadatos preparados, contruimos el contenedor Summarized Experiment:

```r
se <- SummarizedExperiment(
    assays = list(counts = values),
    rowData = features,
    colData = samples,
    metadata = list(title = "Metabotypes of response to bariatric surgery independent
                    of the magnitude of weight loss",
                    repository = "https://github.com/nutrimetabolomics/Metabotyping2018",
                    info = metadata)
)
```

Comprobarmos que el contenedor "Summarized Experiment" este correctamente generado:

```
## class: SummarizedExperiment
## dim: 690 39
## metadata(3): title repository info
## assays(1): counts
## rownames(690): MEDDM_T0 MEDCOL_T0 ... SM.C24.0_T5 SM.C24.1_T5
## rowData names(3): VarName varTpe Description
## colnames(39): 1 2 ... 38 39
## colData names(5): SUBJECTS SURGERY AGE GENDER Group
```

Y comprobamos que los datos y metadatos del contenedor son correctos:

```
## Formal class 'SummarizedExperiment' [package "SummarizedExperiment"] with 5 slots
##   ..@ colData        :Formal class 'DFrame' [package "S4Vectors"] with 6 slots
##   .. .. ..@ rownames       : chr [1:39] "1" "2" "3" "4" ...
##   .. .. ..@ nrows          : int 39
##   .. .. ..@ elementType    : chr "ANY"
##   .. .. ..@ elementMetadata: NULL
##   .. .. ..@ metadata       : list()
##   .. .. ..@ listData       :List of 5
##   .. .. .. ..$ SUBJECTS: num [1:39] 1 2 3 4 5 6 7 8 9 10 ...
##   .. .. .. ..$ SURGERY : chr [1:39] "by pass" "by pass" "by pass" "by pass" ...
##   .. .. .. ..$ AGE     : num [1:39] 27 19 42 37 42 24 33 55 40 47 ...
##   .. .. .. ..$ GENDER  : chr [1:39] "F" "F" "F" "F" ...
##   .. .. .. ..$ Group   : num [1:39] 1 2 1 2 1 2 1 1 1 1 ...
##   ..@ assays         :Formal class 'SimpleAssays' [package "SummarizedExperiment"] with 1 slot
##   .. .. ..@ data:Formal class 'SimpleList' [package "S4Vectors"] with 4 slots
##   .. .. .. .. ..@ listData       :List of 1
##   .. .. .. .. .. ..$ counts: num [1:690, 1:39] 0 0 0 1 85 11.4 2.4 NA NA 151 ...
##   .. .. .. .. .. .. ..- attr(*, "dimnames")=List of 2
##   .. .. .. .. .. .. .. ..$ : chr [1:690] "MEDDM_T0" "MEDCOL_T0" "MEDINF_T0" "MEDHTA_T0" ...
##   .. .. .. .. .. .. .. ..$ : chr [1:39] "1" "2" "3" "4" ...
##   .. .. .. .. ..@ elementType    : chr "ANY"
##   .. .. .. .. ..@ elementMetadata: NULL
##   .. .. .. .. ..@ metadata       : list()
##   ..@ NAMES          : chr [1:690] "MEDDM_T0" "MEDCOL_T0" "MEDINF_T0" "MEDHTA_T0" ...
##   ..@ elementMetadata:Formal class 'DFrame' [package "S4Vectors"] with 6 slots
##   .. .. ..@ rownames       : NULL
##   .. .. ..@ nrows          : int 690
##   .. .. ..@ elementType    : chr "ANY"
##   .. .. ..@ elementMetadata: NULL
##   .. .. ..@ metadata       : list()
```

```
##   .. .. ..@ listData       :List of 3
##   .. .. .. ..$ VarName    : chr [1:690] "MEDDM_T0" "MEDCOL_T0" "MEDINF_T0" "MEDHTA_T0" ...
##   .. .. .. ..$ varTpe     : chr [1:690] "integer" "integer" "integer" "integer" ...
##   .. .. .. ..$ Description: chr [1:690] "dataDesc" "dataDesc" "dataDesc" "dataDesc" ...
##   ..@ metadata       :List of 3
##   .. ..$ title     : chr "Metabotypes of response to bariatric surgery independent \n
##   .. ..$ repository: chr "https://github.com/nutrimetabolomics/Metabotyping2018"
##   .. ..$ info      : chr [1:10] "Data used in the paper \"Metabotypes of response to bariatric surge
```

Visualizamos parte de la matriz de valores:

```
##              1    2     3    4     5     6    7    8     9    10   11    12
## MEDDM_T0   0.0  0.0  0.00  0.0  0.00  0.00  0.0  0.0  0.00  0.00  0.0   0.0
## MEDCOL_T0  0.0  0.0  0.00  0.0  0.00  0.00  0.0  0.0  0.00  0.00  0.0   0.0
## MEDINF_T0  0.0  0.0  0.00  0.0  0.00  0.00  0.0  1.0  0.00  0.00  1.0   0.0
## MEDHTA_T0  1.0  0.0  0.00  0.0  0.00  0.00  0.0  0.0  0.00  0.00  0.0   0.0
## GLU_T0    85.0 78.0 75.00 71.0 82.00 71.00 80.0 90.0 92.00 84.00 75.0 108.0
## INS_T0    11.4 12.1  8.41 12.8  6.01  9.88  9.2  3.4  5.43  6.98 13.3  16.8
##               13    14    15  16    17    18    19  20  21    22    23  24    25
## MEDDM_T0     0.0   0.0   0.0   0   0.0   0.0   0.0   0   0   0.0   0.0   0   0.0
## MEDCOL_T0    0.0   0.0   0.0   0   0.0   0.0   0.0   0   0   0.0   0.0   0   0.0
## MEDINF_T0    0.0   0.0   0.0   0   0.0   0.0   0.0   1   0   0.0   0.0   1   0.0
## MEDHTA_T0    1.0   1.0   1.0   1   0.0   0.0   0.0   0   0   0.0   0.0   0   0.0
## GLU_T0     101.0 105.0 139.0 106 159.0 103.0 106.0 107 127 111.0 141.0 100 100.0
## INS_T0      17.1  21.3  36.6  20  17.6  29.5  13.3  15  15  12.2  32.3  16  12.8
##               26    27    28    29  30  31    32    33    34    35    36  37
## MEDDM_T0      NA   0.0   0.0   0.0   0   0   0.0   0.0   0.0   0.0   0.0   0
## MEDCOL_T0     NA   0.0   1.0   0.0   0   0   0.0   0.0   0.0   0.0   0.0   0
## MEDINF_T0     NA   0.0   0.0   0.0   0   0   0.0   0.0   0.0   0.0   0.0   1
## MEDHTA_T0     NA   0.0   1.0   0.0   0   0   0.0   1.0   1.0   1.0   0.0   0
## GLU_T0     100.0 100.0 117.0 100.0 263 115 108.0 114.0 101.0 108.0 106.0 115
## INS_T0      11.1  19.6  11.6  13.7  21  19  23.1  27.8  23.7  17.7  16.1  43
##               38    39
## MEDDM_T0     0.0   0.0
## MEDCOL_T0    0.0   0.0
## MEDINF_T0    0.0   0.0
## MEDHTA_T0    0.0   0.0
## GLU_T0     102.0 108.0
## INS_T0      21.9  42.7
```

Visualizamos parte del metadata de las filas (características):

```
## DataFrame with 690 rows and 3 columns
##                  VarName      varTpe Description
##              <character> <character> <character>
## MEDDM_T0       MEDDM_T0     integer    dataDesc
## MEDCOL_T0     MEDCOL_T0     integer    dataDesc
## MEDINF_T0     MEDINF_T0     integer    dataDesc
## MEDHTA_T0     MEDHTA_T0     integer    dataDesc
## GLU_T0           GLU_T0     integer    dataDesc
## ...                 ...         ...         ...
## SM.C18.0_T5 SM.C18.0_T5     numeric    dataDesc
## SM.C18.1_T5 SM.C18.1_T5     numeric    dataDesc
```

```
## SM.C20.2_T5 SM.C20.2_T5     numeric    dataDesc
## SM.C24.0_T5 SM.C24.0_T5     numeric    dataDesc
## SM.C24.1_T5 SM.C24.1_T5     numeric    dataDesc
```

Visualizamos parte del metadata de las columnas (muestras):

```
## DataFrame with 39 rows and 5 columns
##      SUBJECTS     SURGERY       AGE      GENDER     Group
##     <numeric> <character> <numeric> <character> <numeric>
## 1           1     by pass        27           F         1
## 2           2     by pass        19           F         2
## 3           3     by pass        42           F         1
## 4           4     by pass        37           F         2
## 5           5     tubular        42           F         1
## ...       ...         ...       ...         ...       ...
## 35         35     tubular        39           M         2
## 36         36     tubular        35           M         1
## 37         37     by pass        46           M         2
## 38         38     tubular        41           M         1
## 39         39     by pass        26           M         1
```

```
## $title
## [1] "Metabotypes of response to bariatric surgery independent \n                    of the magnitude
##
## $repository
## [1] "https://github.com/nutrimetabolomics/Metabotyping2018"
##
## $info
##  [1] "Data used in the paper \"Metabotypes of response to bariatric surgery independent of the magni
##  [2] ""
##  [3] "The data has been published in the paper web site and it also has an independent repository [h
##  [4] ""
##  [5] "The dataset is formed by the following files :"
##  [6] ""
##  [7] "- DataInfo_S013.csv: Metadata. Information on each column in the \"DataValues_S013.csv\" file.
##  [8] "- DataValues_S013.csv: Clinical and metabolomic values for 39 patients at 5 time points."
##  [9] "- AAInformation_S006.csv: Additional information on metabolites in the \"DataValues_S013.csv\"
## [10] ""
```

## Exploración de los datos

### Análisis estadístico univariante

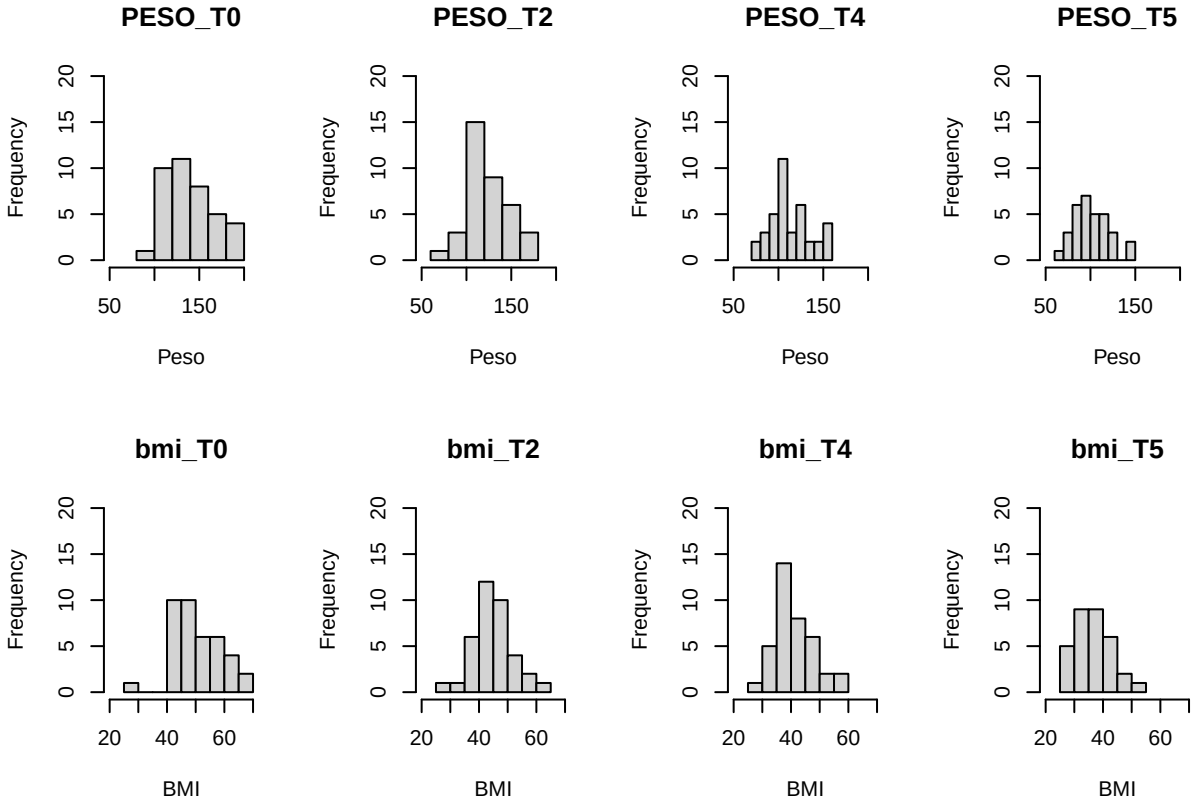
Para empezar a explorar los datos podemos llevar a cabo un análisis estadístico de las variables de nuestro dataset. Dado que tenemos muchas variables, y se trata de un estudio con pacientes con cirugái bariátrica, vamos a centramos en las variables Peso y BMI (body mass index) en los diferentes tiempos.

```
##     PESO_T0         PESO_T2         PESO_T4          PESO_T5
##  Min.   : 84.0   Min.   : 75.7   Min.   : 74.50   Min.   : 64.00
##  1st Qu.:119.5   1st Qu.:110.0   1st Qu.: 99.83   1st Qu.: 88.42
##  Median :135.0   Median :120.0   Median :109.50   Median :100.00
##  Mean   :140.0   Mean   :124.5   Mean   :113.99   Mean   :100.41
```

```
##  3rd Qu.:155.0   3rd Qu.:140.0   3rd Qu.:126.75   3rd Qu.:112.25
##  Max.   :200.0   Max.   :176.0   Max.   :157.00   Max.   :142.00
##                  NA's   :2       NA's   :1        NA's   :7
```

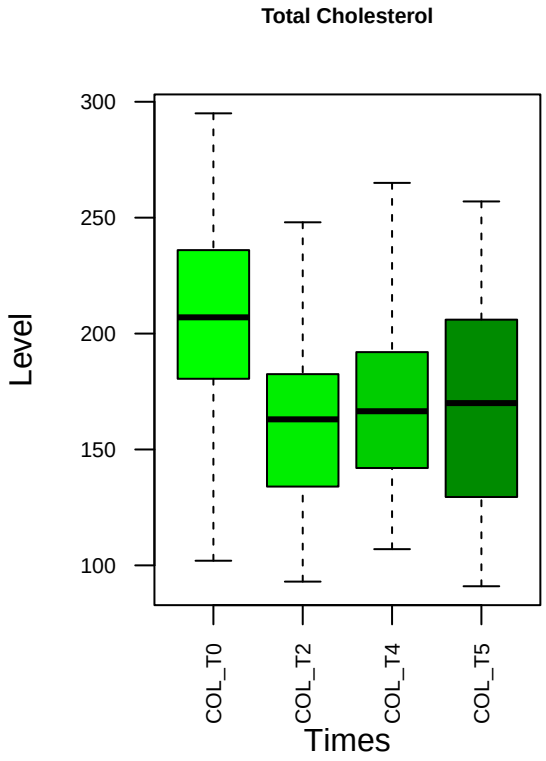
```
##      bmi_T0          bmi_T2          bmi_T4          bmi_T5
##  Min.   :29.80   Min.   :26.80   Min.   :26.40   Min.   :25.50
##  1st Qu.:44.40   1st Qu.:40.20   1st Qu.:36.80   1st Qu.:32.55
##  Median :48.80   Median :44.60   Median :40.00   Median :35.53
##  Mean   :50.52   Mean   :45.09   Mean   :41.28   Mean   :36.42
##  3rd Qu.:55.35   3rd Qu.:49.00   3rd Qu.:44.90   3rd Qu.:40.58
##  Max.   :68.60   Max.   :60.70   Max.   :57.40   Max.   :50.90
##                  NA's   :2       NA's   :1       NA's   :7
```

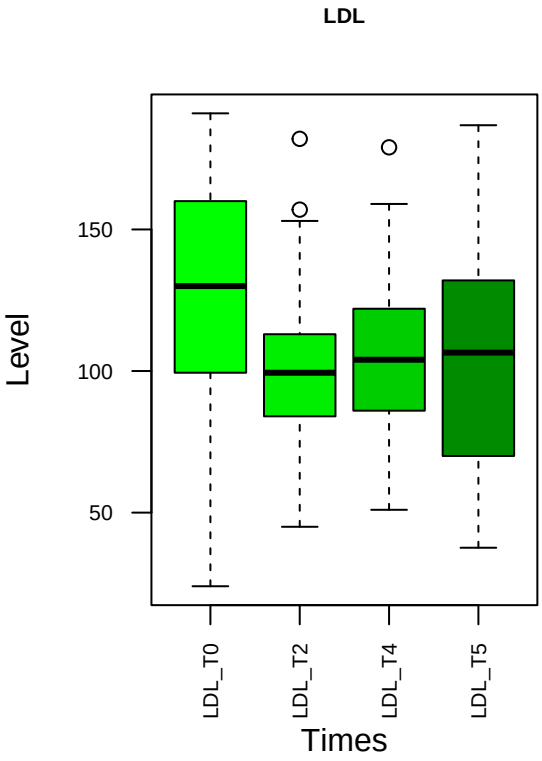
Obteniendo un resumen estadistico de las diferentes variables, parece que hay una reducción progresiva del peso y BMI tras la operación bariátrica. No osbtante, para visualizar estos datos podemos representarlos en histogramas:

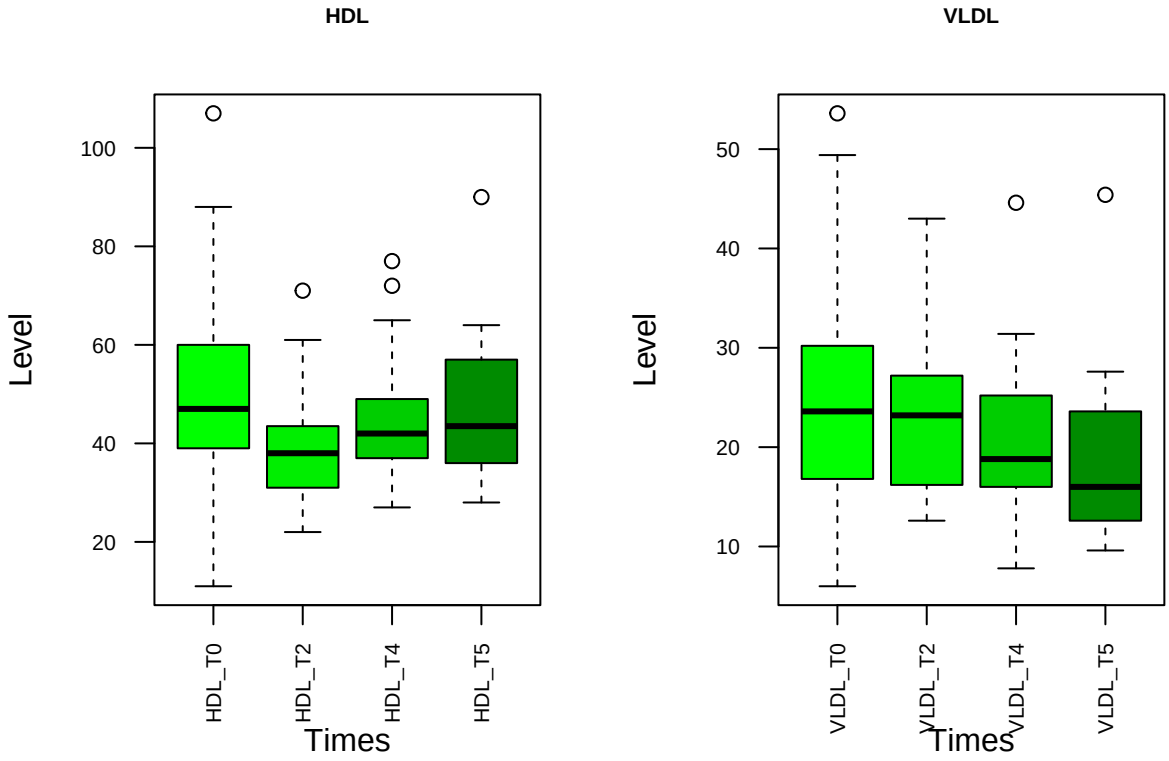



Como se puede observar comparando los histogramas, tras la cirugía hay una bajada de peso y de BMI que se hace envidente entre los tiempos de medición.

Además, podemos explorar otras variables, como son el Colesterol Total, LDL, HDL y VLDL, las cuales podrían verse afectadas por la cirugía bariátrica debido a la relación que guardan con el tejido adiposo. En este caso, para ver el cambio que sufren estos metabolitos con el tiempo representamos diagramas de caja y bigotes o boxplots.


Total Cholesterol
Level
300
250
200
150
100
COL_T0
COL_T2
COL_T4
COL_T5
Times

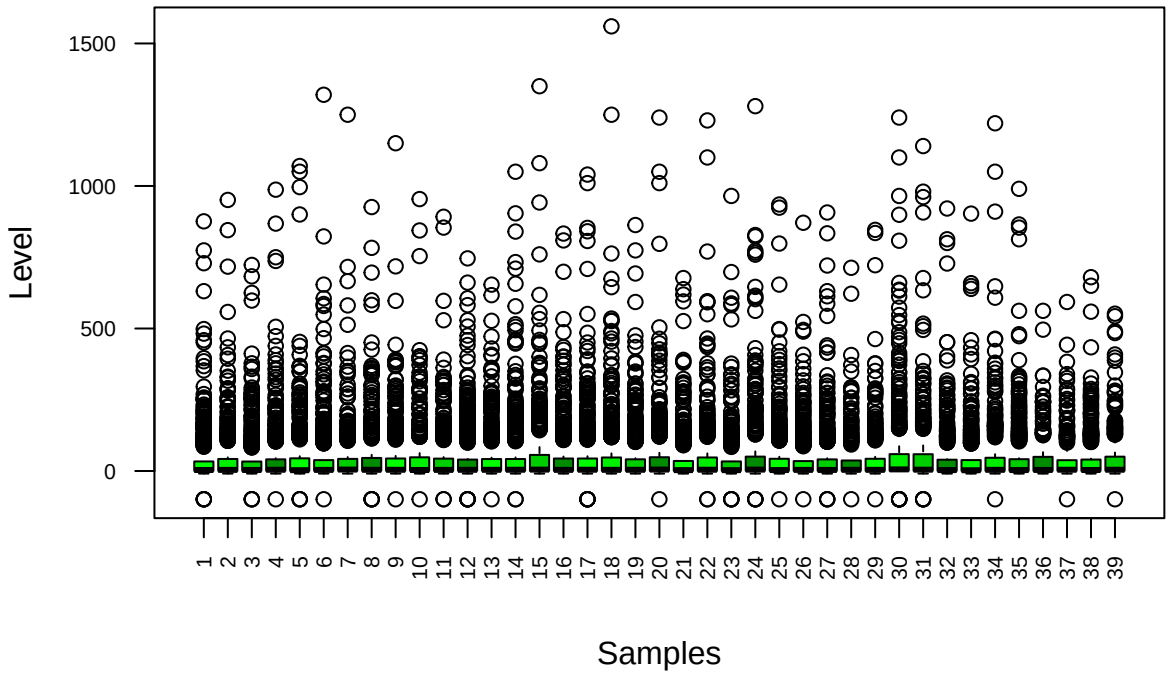

LDL
Level
150
100
50
LDL_T0
LDL_T2
LDL_T4
LDL_T5
Times



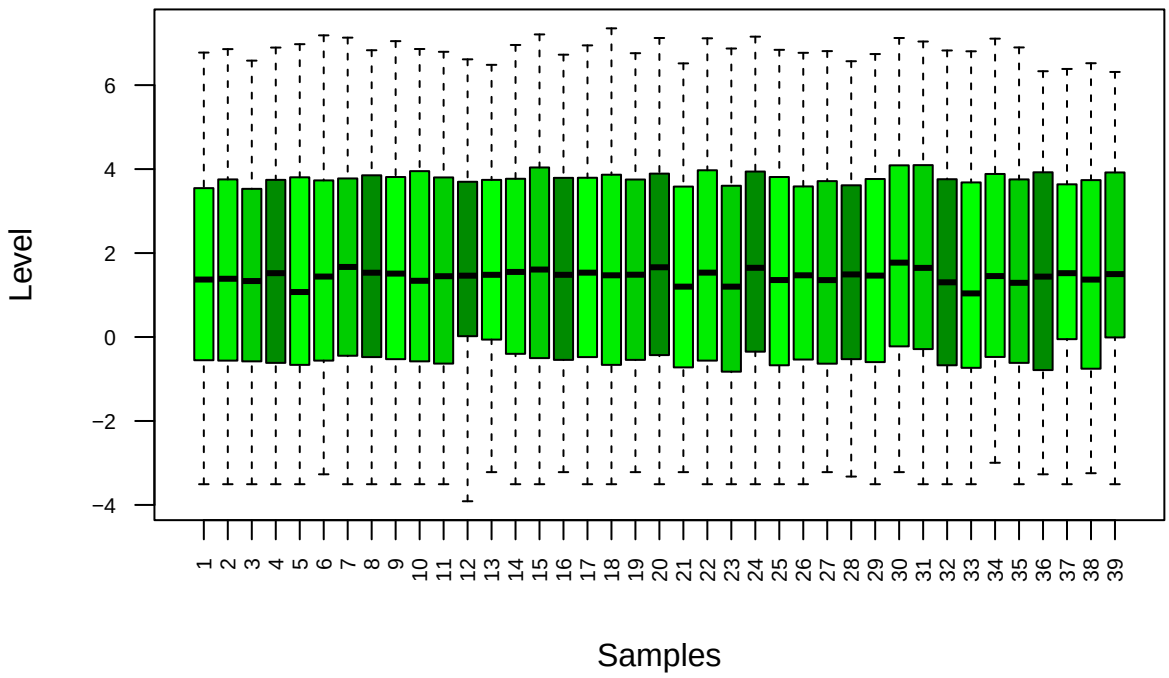

Podemos observar que tanto el Colesterol total, como el LDL y HDL bajan significantivante tras las operación bariátrica, y luego presentan una pequeña recuperación con el tiempo. Por otro lado, el VLDL no presenta una bajada drástica tras la operación, sino que los niveles se van reduciendo progresivamente con el timpo. Por lo tanto, los datos parecen indicar que la operación bariátrica ha tenido un efecto sobre los niveles de los distintos tipos de colesterol.

### Visualización de los datos multivariante

Para resalizar una exploración multivariante de los datos, en primer lugar representamos los valores de las muestras en boxplots, para determinar si es combeniente realizar alguna clase de preprocesamiento.




Dada la asimetría que presentan los datos, lo corregimos tomamos logaritmos de los valores.

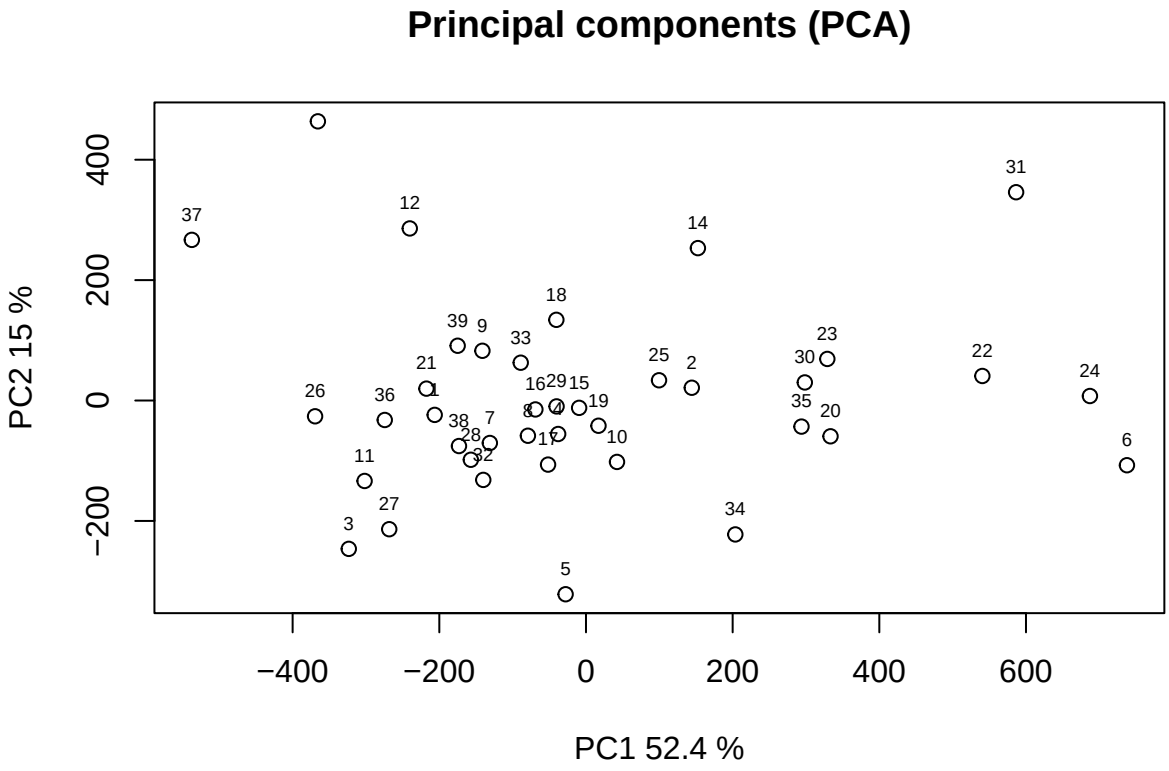



En la represntación de los valores, las diferencias entre las muestras no son extremadamente grandes, por lo que no se espera que afecte considerablemente a los resultados. De esta forma, no se va a aplicar un preprocesado o corrección de los datos para los pasos posteriores.

A continuación, llevamos a cabo un Análisis de Componentes Principales (PCA), que nos permite determinar factores latentes o un effecto batch en las muestras. Sin embargo, existen variables para los cuales tenemos valores faltantes (NAs) en algunas muestras. Dado que no podemos calcular las componentes principales utilizando dichas variables, filtramos la matriz de valores eliminando dichas filas con la función na.omit().

Una vez comprobado que no hay NAs (ni otros valores no finitos) en las variables restantes tas el filtrado, calculamos las componentes principales con la función prcomp(). Representamos las dos primeras componentes en un gráfico.

```
PC_se <- prcomp(t(counts_filtered), scale= FALSE)
```




La primera componente explica el 52.4% de la variabilidad, mientras que la segunda componente explica el 15%. Por tanto, las dos primeras componentes son capaces de explicar el 67.4% de la variabilidad de las muestras.

En el caso de la PC1, podemos obtener los coeficientes para las variables con mayor peso:

```
##           Gln_T0    PC.aa.C34.2_T0           Ala_T0           Lys_T0
##        0.72544799        0.32702119        0.27470305        0.24127975
##           Pro_T0           Val_T0           Gly_T0    PC.aa.C36.2_T0
##        0.21510583        0.20092794        0.13494218        0.13095333
##           Thr_T0    PC.aa.C34.1_T0           Leu_T0    PC.aa.C36.4_T0
##        0.11260267        0.10137622        0.09812576        0.09023839
##    PC.aa.C38.4_T0    PC.aa.C36.3_T0           Orn_T0           Ser_T0
##        0.08200174        0.07739634        0.07451676        0.06973422
##           Tyr_T0           His_T0 lysoPC.a.C16.0_T0      SM.C16.0_T0
##        0.06462855        0.06349171        0.05741743        0.05182122
```

Se puede observar que la Glutamina en el tiempo O (Gln_T0) es la que tiene mayor peso en la PC1, con un coeficiente de 0.72. De esta forma, las muestras que se encuentran más a la derecha del gráfico poseen niveles superiores de Glutamina en el tiempo 0 (Gln_T0).

Además, las variables correspondientes a los niveles de aminoacidos en el tiempo 0 son algunas de las que más peso tienen en la PC1, por tanto se podría interpretar como un factor **Niveles de Aninoácidos**.

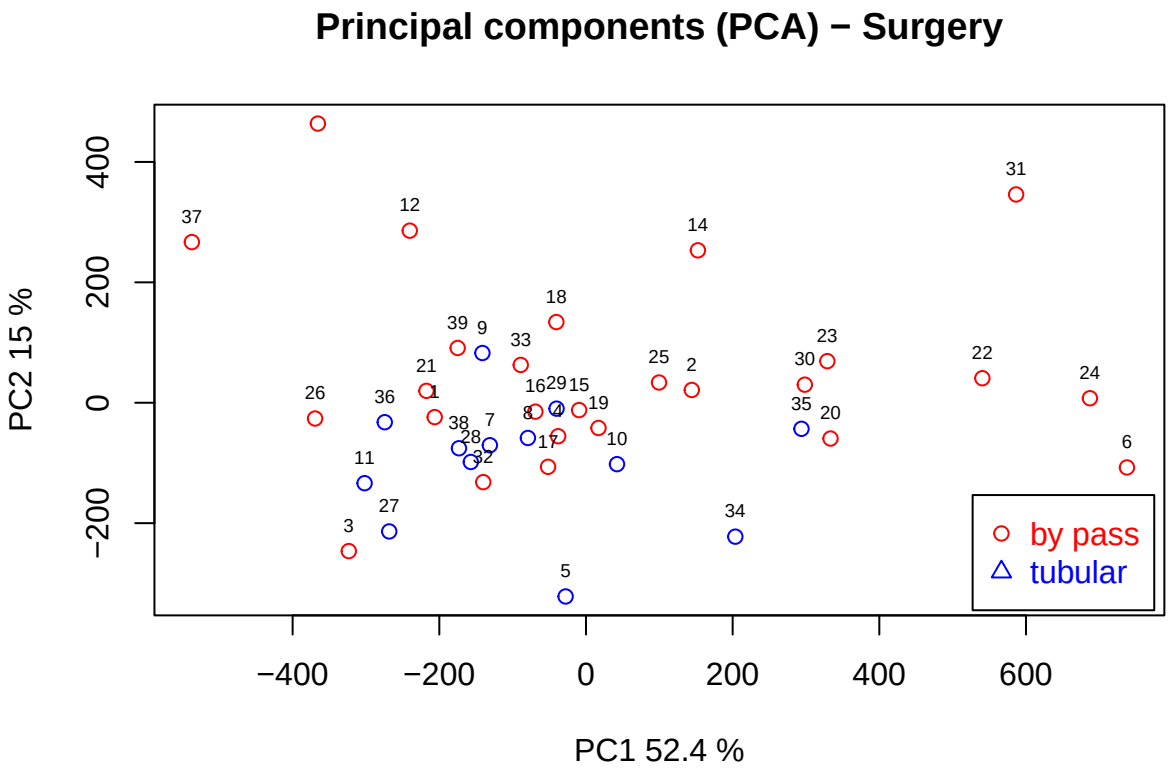
Realizamos lo mismo con la PC2:

```
##           Ala_T0           Glu_T0    PC.aa.C34.2_T0           TG_T0
```

```
##        0.52755178        0.28802613        0.27469409        0.21700975
## lysoPC.a.C16.0_T0           Val_T0    PC.aa.C34.1_T0           Arg_T0
##        0.19545299        0.18071141        0.15217974        0.13471944
##           Pro_T0           Lys_T0           Leu_T0    PC.aa.C36.3_T0
##        0.12794393        0.12046886        0.10329764        0.09035260
##    PC.aa.C36.2_T0           COL_T0 lysoPC.a.C18.0_T0           Ile_T0
##        0.08242986        0.07323029        0.06860115        0.06823328
##           Phe_T0    PC.aa.C38.3_T0           Asp_T0    PC.aa.C38.6_T0
##        0.05923462        0.05860880        0.04693310        0.04597488
```

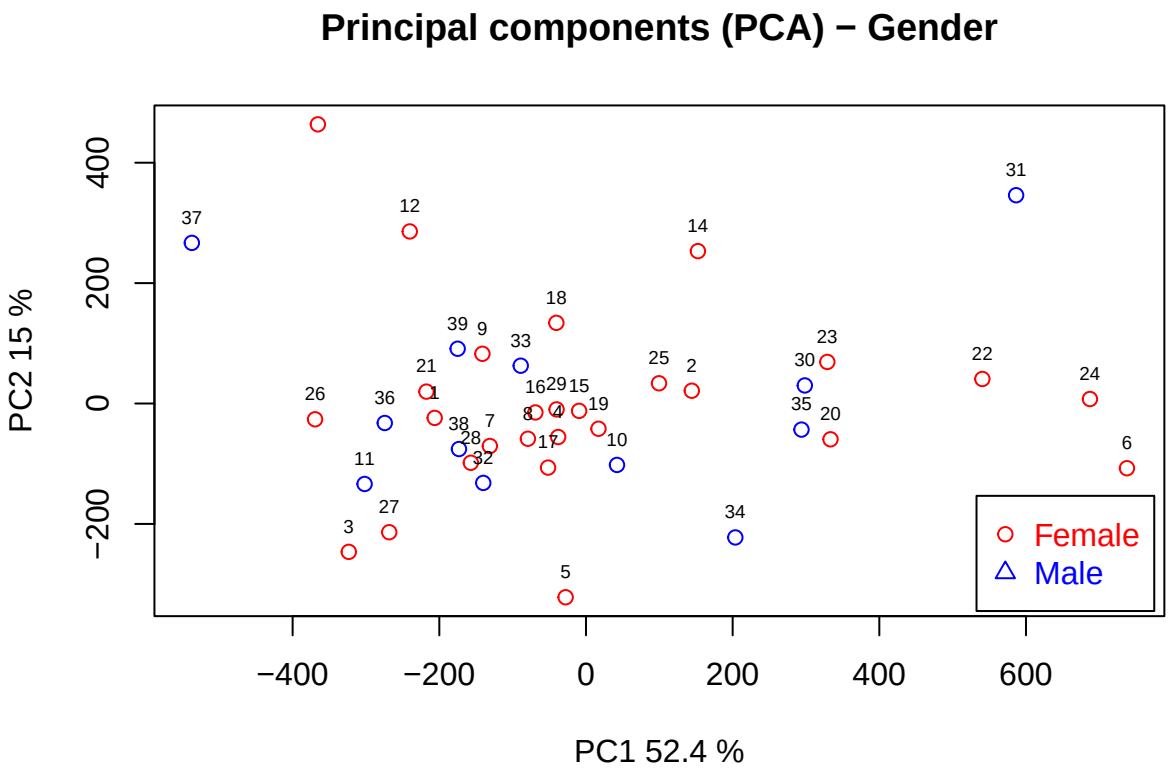
Podemos observar que con la PC2 ocurre algo parecido a la componente principal 1, donde los niveles de aminoacidos en el tiempo 0 son algunas de las variables con mayor peso. No obstante, en la PC2 es la Alanina en el tiempo 0 la que tiene un mayor peso, con un coeficiente de 0.52.

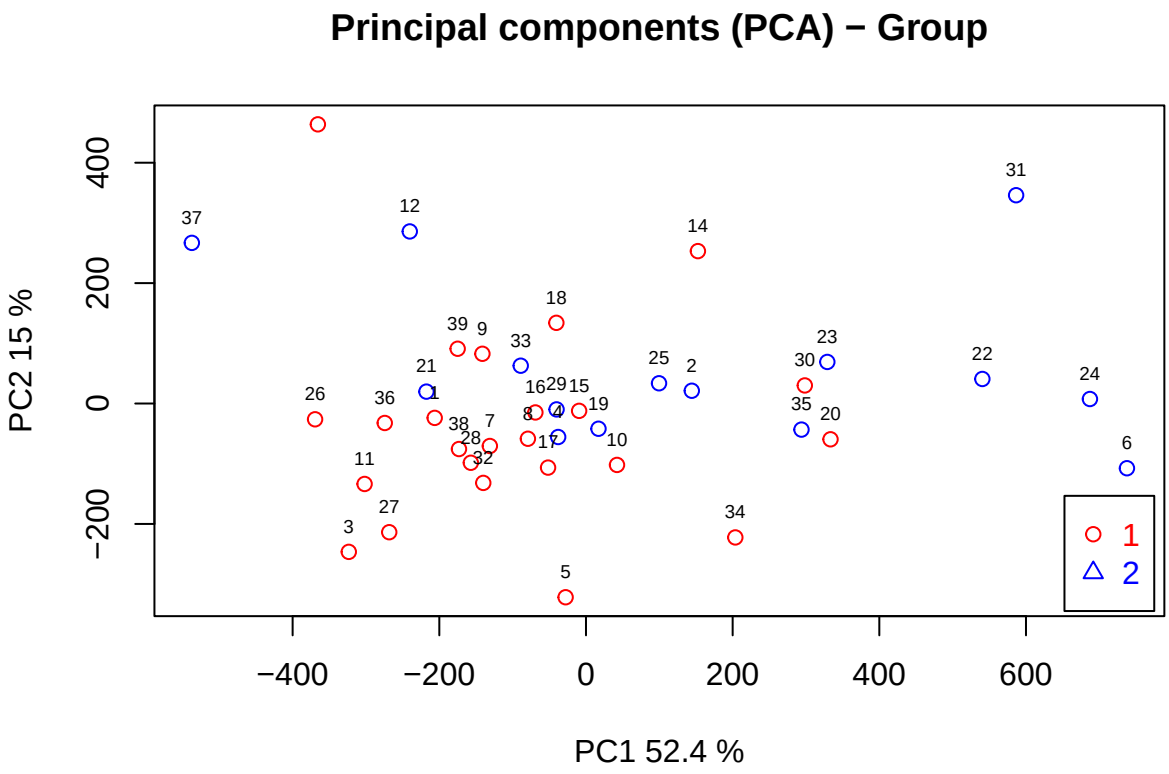
Por otro lado, con el PCA podemos determinar si existe algun effecto batch en base al timpo de cirugía empleado:




Aparentemente, no existe ningún efecto batch claro, aunque las muestras con cirugía por by pass parecen estar un poco por encima de las muestras con cirugía tubular.

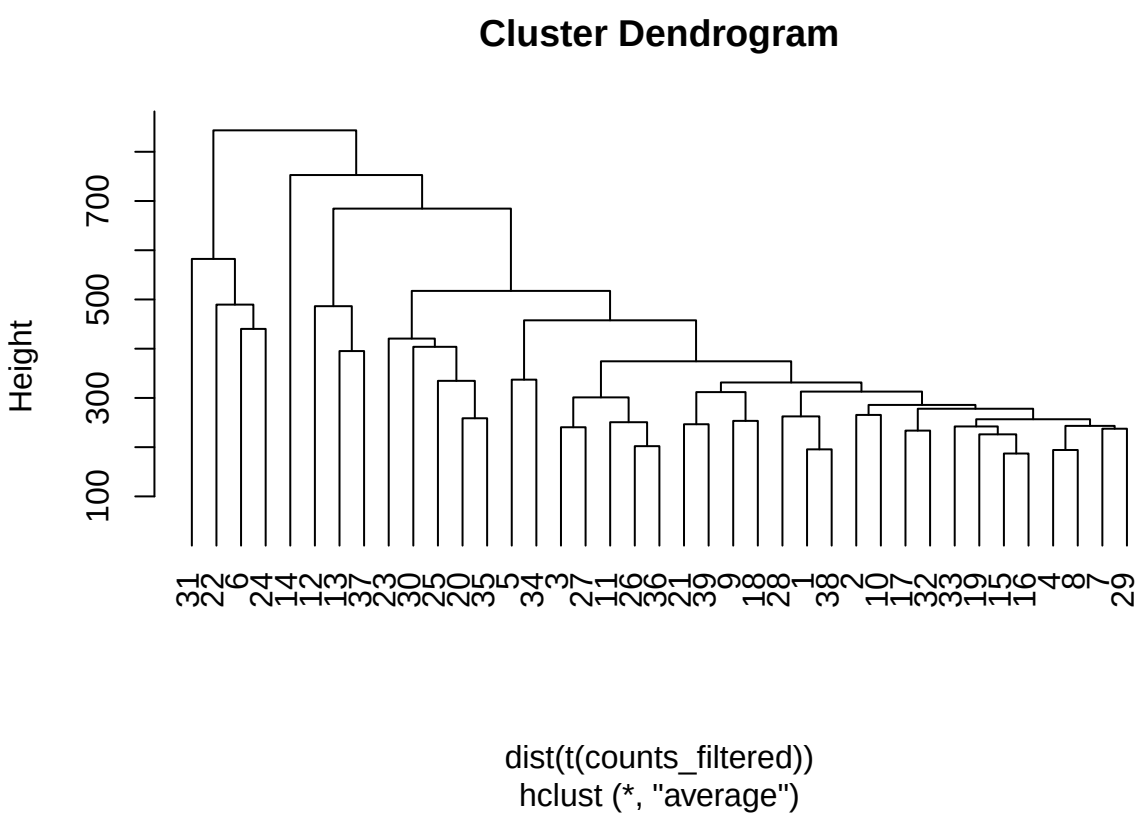
Además, tambíen podemos observar si hay diferencias en el PCA en base al genero o el grupo:


Principal components (PCA) − Gender
PC2 15 %
PC1 52.4 %
−400
−200
0
200
400
600
−200
0
200
400
Female
Male

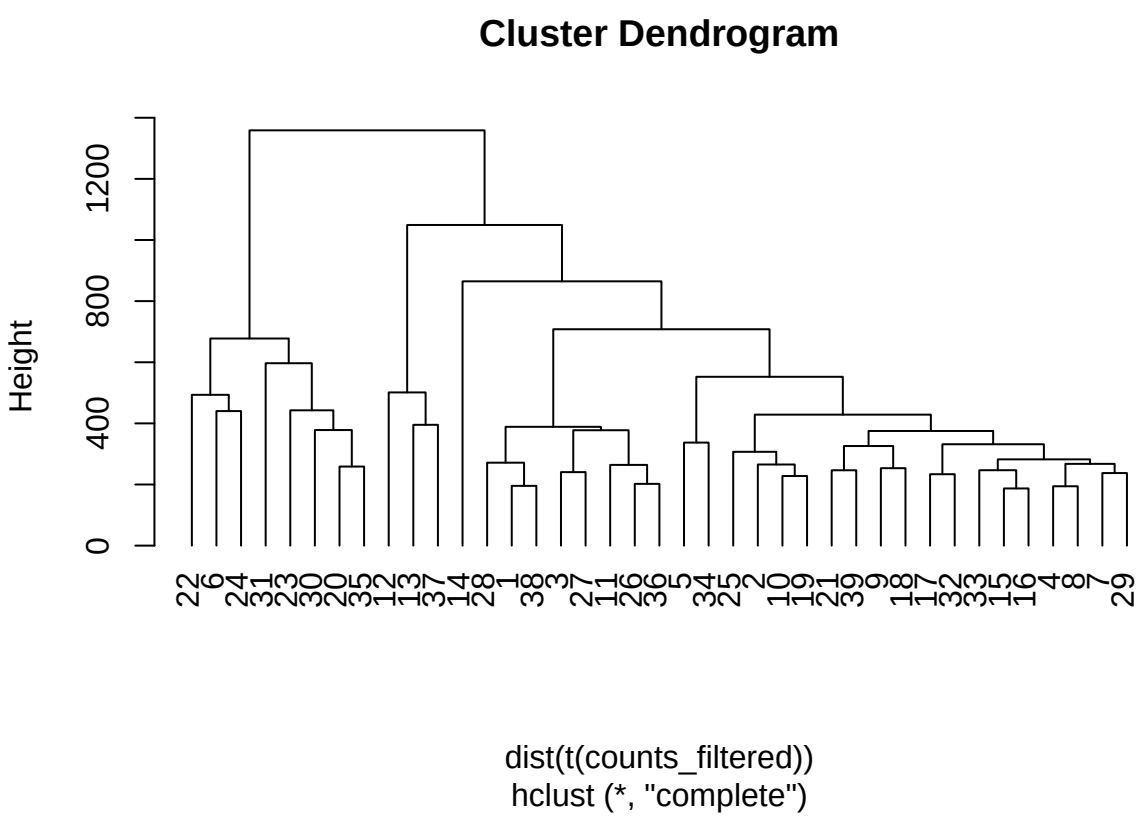



Como se puede observar, no hay diferencias aparantes en la representación de las componentes principales en base al genero o al grupo.

Una forma de estudiar la agrupación de las muestras es mediante un agrupación jerárquica. Para realizar esta representación utilizamos la función hclust() con el método de aglomeración “average”.




No se distinguen grupos o cluster claros. No obstante, se pueden utilizar otros métodos de aglomeración, como por ejemplo el método “complete” que es el método por defecto de la función hclust().




Con este método se pueden distinguir dos agrupaciones o clusters. Un primer cluster formado por 8 muestras donde predomina la cirugía por "by pass", y un segundo cluster formado por el resto de muestras. Además, en el PCA el primer cluster se encuentra a la derecha y por tanto estan definidos por valores elevados de glutamina (Gln), de acuerdo a la contribución que tiene dicha variable en la Componente Principal 1.

### Reposición de los datos en Github

Para subir los archivos en un respoitorio de Github en primer lugar creamos una cuenta en Github. Luego le damos a "New Repository" para crear un nuevo repositorio. Y por ultimo subimos los archivos con a dicho respositorio

En el respositorio se deben de incluir una serie de archivos generados a partir de este dataset y la exploración de los datos.

Uno de estos archivos es el objeto contenedor con los datos y metadatos en formato binario (.Rda), por lo que guardamos este objeto en dicho formato.

También necesitamos los datos en formato texto, por lo que guardamos la matriz de valores del objeto contenedor en formato .csv.

los metadatos se covirtieron en formato markdown y luego se creo un archivo markdown (.md) con la función writeLines().

Además, el código R para la exploración de datos se copió en un archivo R script.

Todos estos archivos, junto con el presente informe, fueron subidos en un repositorio de github denominado "Gancedo-Verdejo-Javier-PEC1", con dirección URL:

https://github.com/jgancedov/Gancedo-Verdejo-Javier-PEC1.git

## Discusión, limitaciones y conclusiones del estudio

El presente estudio ha relevado algunos hallazgos interesantes sobre los datos metabólicos de este dataset. De hecho, se han observado cambios en algunos parámetros y metabolitos tras la cirugía bariátrica. En el caso del peso y el BMI se ha observado una clara tendencia a la reducción de estos tras la cirugía, destacando la eficiencia e impacto de esta práctica clínica. Además, se han observado reducciones de los niveles de colesterol total, LDL y HDL justo después de la cirugía, esto confirma que la operación bariátrica tiene un efecto relevante sobre el metabolismo de los lípidos. Por otro lado, el análisis multivariante indicó que los niveles en el momento de la operación de algunos aminoácidos (sobre todo la glutamina y la alanina) eran los principales contribuyentes a la variabilidad de los datos.

Con el "clustering" o agrupación jerárquica, y junto con el PCA, se identificaron dos grupos de muestras, caracterizados por niveles diferentes de glutamina. Además, también es posible dicha separación de grupos estuviera influenciada por el tipo de cirugía. Sin embargo, esta exploración de los datos no permite determinar si existe o no una influencia clara, por lo que sería necesario realizar otro tipo de análisis para determinar si efectivamente existe este efecto del tipo de cirugía sobre nuestros datos, y si por tanto sería necesario eliminar este efecto batch para los análisis posteriores.

Este estudio presenta varias limitaciones que hay que tener presentes. En primer lugar, el tamaño muestral podría ser escaso para interpretar y concluir correctamente los análisis y resultados, por lo que un tamaño mayor seria recomendable. Por otra parte, hay una alta presencia de valores ausentes lo que repercute en tener que despreciar muchas variables en los análisis multivariantes, por lo que se esta perdiendo mucha información dificultando la interpretación de los resultados. Además, los tiempos en los que se tomaron las medidas no son tiempos consecutivos, y por tanto cabria esperar que los intervalos de tiempo no sean siempre iguales, lo cual dificulta en gran medida estudiar la progresión de los parámetros y metabolitos en el tiempo tras la cirugía.

En conclusión, este estudio demuestra la efectividad de la cirugía bariátrica en la reducción de peso y BMI, así como el impacto sobre el colesterol y el metabolismo lipídico. Por otro lado, hemos sido capaces de desentrañar la variabilidad de los datos y la importancia de los aminoácidos en esta. Además, el uso de un contenedor de tipo SummarizedExperiment permite realizar un seguimiento correcto de la información y las muestras, mientras que el uso del repositorio de GitHub asegura la reproducibilidad de los datos.

## Apendice 1 - Código de R

Aquí se añade el código de R utilizado para el análisis presentado en este informe.

```
########################################
## Obtención y preparación de los datos ##
########################################

#Instalar y cargar los paquetes necesarios
if (!require("BiocManager", quietly = TRUE))
  install.packages("BiocManager")
BiocManager::install()

if(!require(SummarizedExperiment)) {
  BiocManager::install("SummarizedExperiment")
}
require(SummarizedExperiment)

library(readr)

#Cargamos los archivos
```

```r
DataInfo_S013 <- read_csv("Datasets/2018-MetabotypingPaper/DataInfo_S013.csv")
DataValues_S013 <- read_csv("Datasets/2018-MetabotypingPaper/DataValues_S013.csv")
head(DataInfo_S013)
head(DataValues_S013)

# Para trabajar con las trablas las transformamos a Dataframes
DataInfo_S013 <- as.data.frame(DataInfo_S013)
DataValues_S013 <- as.data.frame(DataValues_S013)

# Asignamos la primera columna como rownames
rownames(DataInfo_S013) <- DataInfo_S013[,1]
DataInfo_S013[,1] <- NULL

rownames(DataValues_S013) <- DataValues_S013[,1]
DataValues_S013[,1] <- NULL

# Extraemos la información de la matriz de valores
values <- DataValues_S013[,6:695]
values <- t(values)

# Extraemos la información de las columnas o samples
samples <- DataValues_S013[,1:5]

# Extraemos la información de las filas o features
features <- DataInfo_S013[6:695,]

# Cargamos la información del dataset como metadata con la función readLines()
metadata <- readLines("Datasets/2018-MetabotypingPaper/description.md")




#########################################################
## Creación del Contenedor "Summarized Experiment" ##
#########################################################

#contruimos el contenedor Summarized Experiment:
se <- SummarizedExperiment(
  assays = list(counts = values),
  rowData = features,
  colData = samples,
  metadata = list(title = "Metabotypes of response to bariatric surgery independent
                    of the magnitude of weight loss",
                  repository = "https://github.com/nutrimetabolomics/Metabotyping2018",
                  info = metadata)
)

#Comprobarmos que el contenedor "Summarized Experiment" este correctamente generado:
se
str(se)

# Visualizamos parte de los datos y metadatos
head(assays(se)$counts) #Matriz de valores
```

```r
rowData(se) #filas (características)
colData(se) #columnas (muestras)
metadata(se) #metadatos




##############################
## Exploración de los datos ##
##############################

# Resumenes estadísticos (de las variables peso y bmi)
row_weight <- which(grepl(paste("PESO", collapse = "|"), rownames(se)))
summary(t(assays(se)$counts [row_weight,]))

row_bmi <- which(grepl(paste("bmi", collapse = "|"), rownames(se)))
summary(t(assays(se)$counts [row_bmi,]))

# Representación en histogramas (del las variables peso y bmi)
opt <- par(mfrow=c(2,4))
hist(assays(se)$counts [10,], xlab = "Peso", main = "PESO_T0", xlim = c(50,210), ylim = c(0,20))
hist(assays(se)$counts [182,], xlab = "Peso", main = "PESO_T2", xlim = c(50,210), ylim = c(0,20))
hist(assays(se)$counts [356,], xlab = "Peso", main = "PESO_T4", xlim = c(50,210), ylim = c(0,20))
hist(assays(se)$counts [528,], xlab = "Peso", main = "PESO_T5", xlim = c(50,210), ylim = c(0,20))
hist(assays(se)$counts [11,], xlab = "BMI", main = "bmi_T0", xlim = c(20,70), ylim = c(0,20))
hist(assays(se)$counts [183,], xlab = "BMI", main = "bmi_T2", xlim = c(20,70), ylim = c(0,20))
hist(assays(se)$counts [357,], xlab = "BMI", main = "bmi_T4", xlim = c(20,70), ylim = c(0,20))
hist(assays(se)$counts [529,], xlab = "BMI", main = "bmi_T5", xlim = c(20,70), ylim = c(0,20))
par(opt)

# Determinamos las filas que se corresponden a las variables del colesterol
Metabol <- c("COL", "LDL", "HDL", "VLDL")
filas <- which(grepl(paste(Metabol, collapse = "|"), rownames(se)))
filas

# Representación en diagramas de cajas y bigotes (para las variables del colesterol)
groupColors <- c("green","green2","green3","green4")
opt <- par(mfrow=c(1,2))
boxplot(t(assays(se)$counts [c(18,190,364,536),]), col=groupColors, main="Total Cholesterol",
        xlab="Times",
        ylab="Level", las=2, cex.axis=0.7, cex.main=0.7)
boxplot(t(assays(se)$counts [c(19,191,365,537),]), col=groupColors, main="LDL",
        xlab="Times",
        ylab="Level", las=2, cex.axis=0.7, cex.main=0.7)
boxplot(t(assays(se)$counts [c(20,192,366,538),]), col=groupColors, main="HDL",
        xlab="Times",
        ylab="Level", las=2, cex.axis=0.7, cex.main=0.7)
boxplot(t(assays(se)$counts [c(21,193,367,539),]), col=groupColors, main="VLDL",
        xlab="Times",
        ylab="Level", las=2, cex.axis=0.7, cex.main=0.7)
par(opt)

# Representación en diagramas de cajas y bigotes de los valores de las muestras
```

```r
boxplot(assays(se)$counts, col=groupColors,
        xlab="Samples",
        ylab="Level", las=2, cex.axis=0.7, cex.main=0.7)

# Representación en diagramas de cajas y bigotes del logaritmo de los valores de las muestras
boxplot(log(assays(se)$counts), col=groupColors, main=,
        xlab="Samples",
        ylab="Level", las=2, cex.axis=0.7, cex.main=0.7)

#Comprobación y eliminación de variables con NAs
all(is.finite( assays(se)$counts )) #Comprobamos si hay valores no finitos
counts_filtered <- na.omit(assays(se)$counts) #Eliminamos variables con NAs
all(is.finite(counts_filtered)) #Comprobamos si hay valores no finitos

#Analisis de componentes principales (PCA)
PC_se <- prcomp(t(counts_filtered), scale= FALSE)

#Representación del PCA
loads <- round(PC_se$sdev^2/sum(PC_se$sdev^2)*100,1)
xlab<-c(paste("PC1",loads[1],"%"))
ylab<-c(paste("PC2",loads[2],"%"))
plot(PC_se$x[,1:2],xlab=xlab,ylab=ylab,
     main ="Principal components (PCA)")
names2plot<-names(as.data.frame(counts_filtered))
text(PC_se$x[,1],PC_se$x[,2],names2plot, pos=3, cex=.6)

#Coeficientes de las componentes principales 1 y 2
PC1 <- PC_se$rotation[,1]
pC1_sorted <- sort(PC1, decreasing = TRUE)
pC1_sorted[1:20]

PC2 <- PC_se$rotation[,2]
pC2_sorted <- sort(PC2, decreasing = TRUE)
pC2_sorted[1:20]

#Representación del PCA en base a la cirugía
palette <- c("red", "blue")
surgery <- as.factor(colData(se)$SURGERY)
colors_surgery <- palette[as.numeric(surgery)]

plot(PC_se$x[,1:2],xlab=xlab,ylab=ylab, col = colors_surgery,
     main ="Principal components (PCA) - Surgery")
text(PC_se$x[,1],PC_se$x[,2],names2plot, pos=3, cex=.6)
legend( "bottomright"  , inset = c(0.01,0.01), cex =1, bty = "y", legend = c("by pass", "tubular"),
        text.col = c("red", "blue"), col = c("red", "blue"), pt.bg = c("red","blue"), pch = c(1,2))

#Representación del PCA en base al genero
gender <- as.factor(colData(se)$GENDER)
colors_gender <- palette[as.numeric(gender)]

plot(PC_se$x[,1:2],xlab=xlab,ylab=ylab, col = colors_gender,
     main ="Principal components (PCA) - Gender")
text(PC_se$x[,1],PC_se$x[,2],names2plot, pos=3, cex=.6)
```

```
legend( "bottomright"  , inset = c(0.01,0.01), cex =1, bty = "y", legend = c("Female", "Male"),
        text.col = c("red", "blue"), col = c("red", "blue"), pt.bg = c("red","blue"), pch = c(1,2))

#Representación del PCA en base al grupo
group <- as.factor(colData(se)$Group)
colors_group <- palette[as.numeric(group)]

plot(PC_se$x[,1:2],xlab=xlab,ylab=ylab, col = colors_group,
     main ="Principal components (PCA) - Group")
text(PC_se$x[,1],PC_se$x[,2],names2plot, pos=3, cex=.6)
legend( "bottomright"  , inset = c(0.01,0.01), cex =1, bty = "y", legend = c("1", "2"),
        text.col = c("red", "blue"), col = c("red", "blue"), pt.bg = c("red","blue"), pch = c(1,2))

#Agrupación jerárquica con el método average y representación
clust.euclid.average <- hclust(dist(t(counts_filtered)),method="average")
plot(clust.euclid.average, hang=-1)

#Agrupación jerárquica con el método complete y representación
clust.euclid.complete <- hclust(dist(t(counts_filtered)),method="complete")
plot(clust.euclid.complete, hang=-1)




########################################
## Reposición de los datos en Github ##
########################################

# Guardamos el objeto contenedor se en un archivo .Rda
save(se, file = "Contenedor_SummarizedExperiment.Rda")

#guardamos la matriz de valores del objeto contenedor en formato .csv
write.csv(assays(se)$counts, file = "Datos.csv", row.names = TRUE)

# Convertir los metadatos del dataset, de las filas y columnas a formato markdown
metadata_dataset <- knitr::kable(metadata(se), format = "markdown")
metadatos_samples_md <- knitr::kable(as.data.frame(colData(se)), format = "markdown")
metadatos_features_md <- knitr::kable(as.data.frame(rowData(se)), format = "markdown")
# Crear el archivo markdown
writeLines(c("# Metadatos del Dataset", metadata_dataset, "# Metadatos de columnas",
             metadatos_samples_md, "# Metadatos de filas", metadatos_features_md), "metadatos.md")
```