scrip2clase.R

arturo

2023-05-11

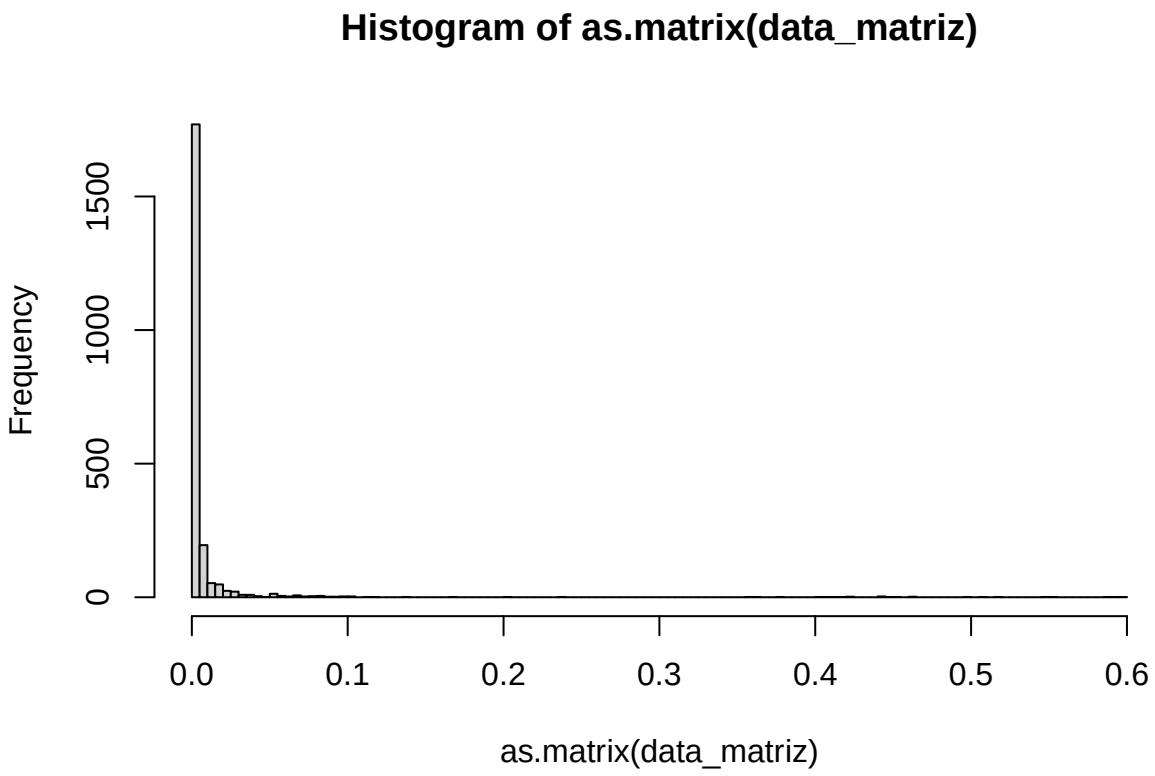
```
##EJEMPLO ANALISIS DE DATOS GENERADOS POR GCMS###################
##MC.Arturo Ramírez-Ordorica curso 2023
library(ggplot2)
library(viridis)
```

```
## Loading required package: viridisLite
```

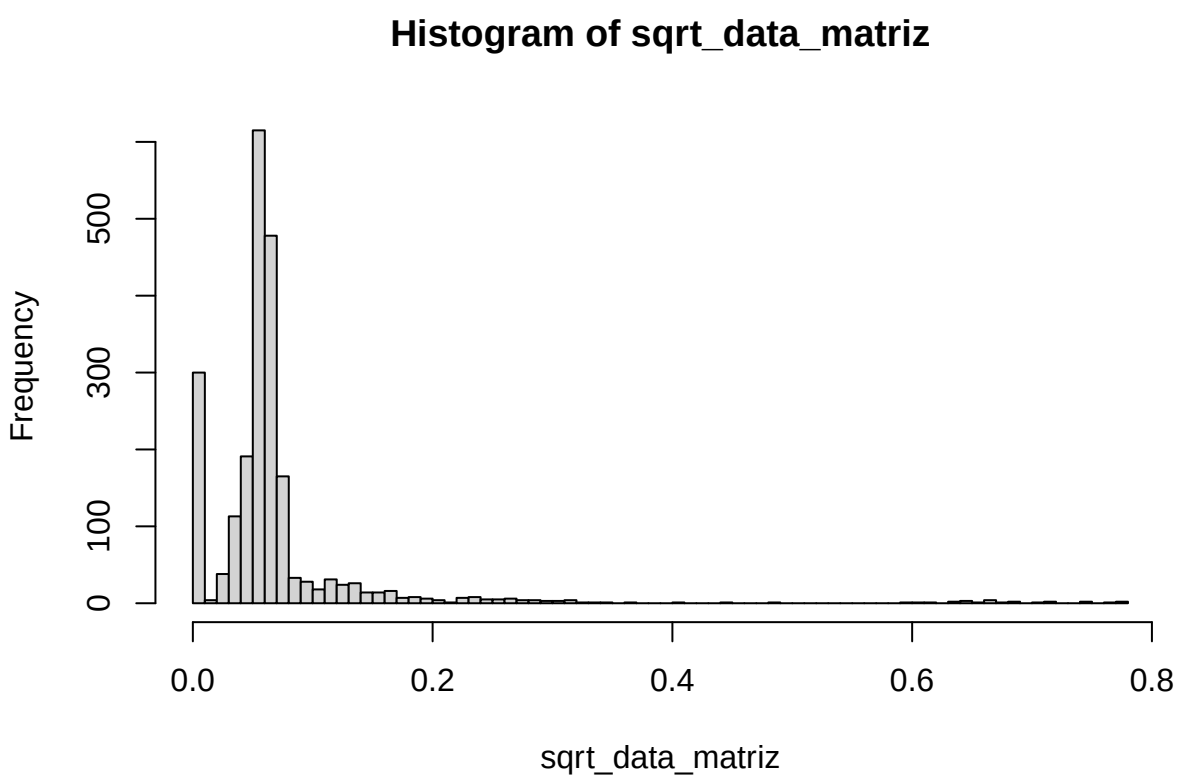
```
# Cargar los datos
data <- read.csv("tricopca.csv",header = T)
head(names(data),n=10)
```

```
##  [1] "Tratamiento"                       "Fuente"
##  [3] "Alcohol.isopropilico"              "Etanol"
##  [5] "X2.5.Dimetilfurano"                "Acetic.acid..2.methylpropyl.ester"
##  [7] "X1.Propanol"                       "Dimetil.disulfuro"
##  [9] "X2.Metil.propanol"                 "X3.Metil.butanol"
```

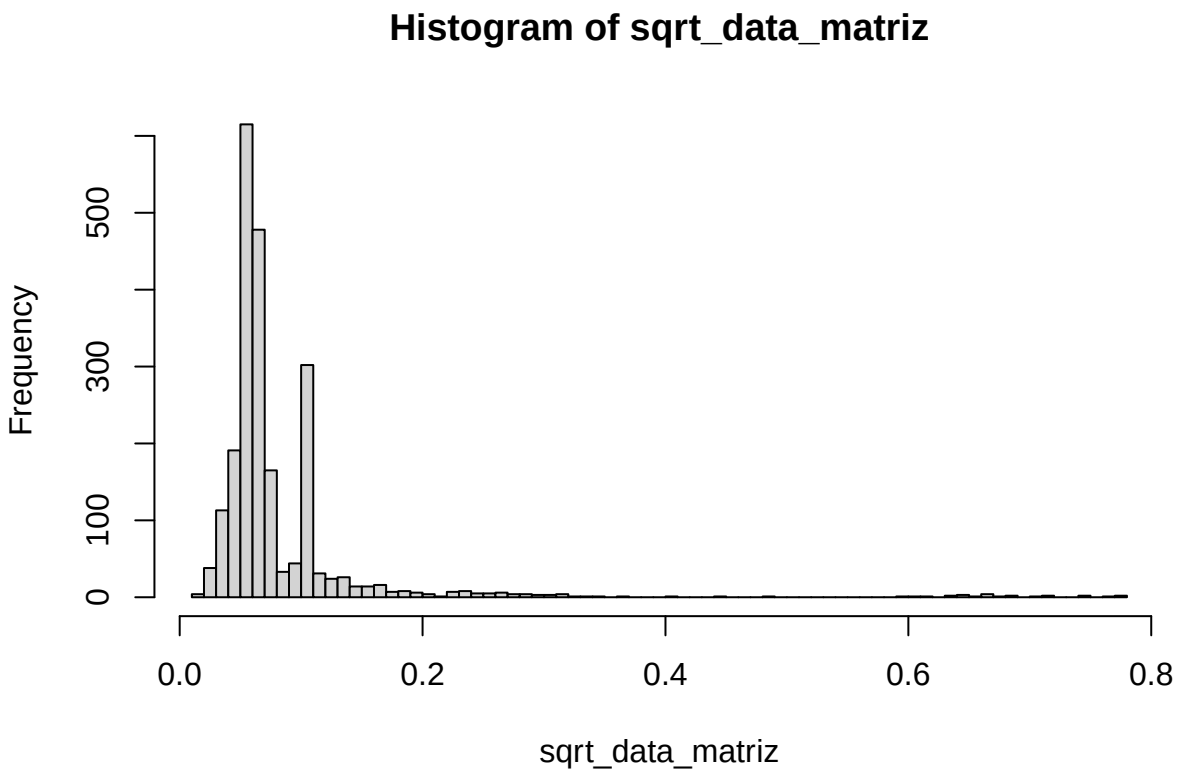
```
#Preparar los datos
data_matriz<-as.matrix(data[,-c(1:2)])

#Graficar intensidades por cada tratamiento
hist(as.matrix(data_matriz),breaks=100)
```



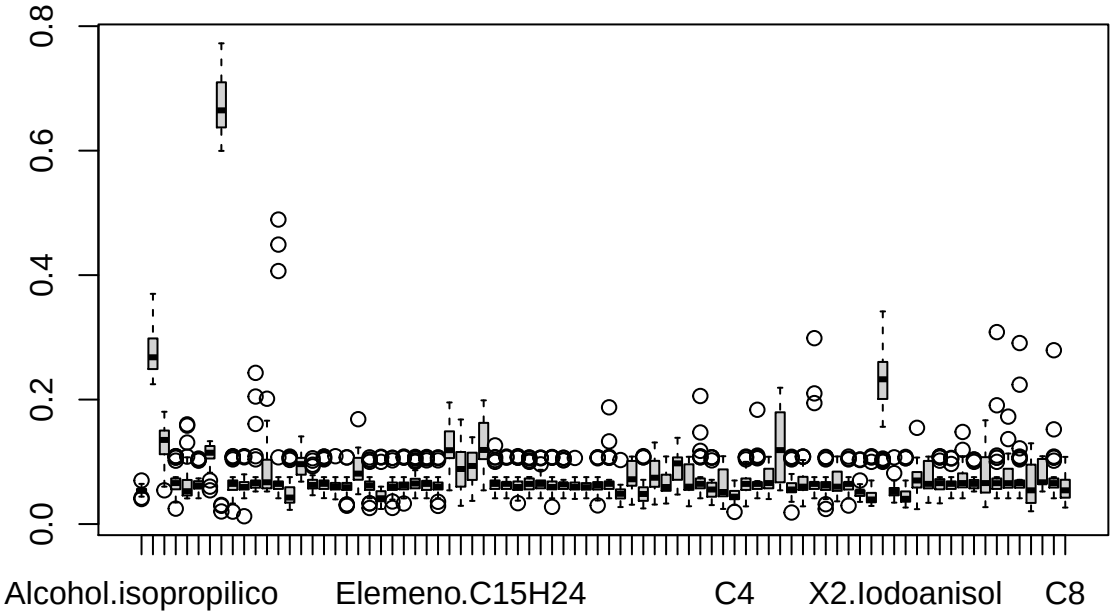

```
sqrt_data_matriz<-sqrt(as.matrix(data_matriz))
hist(sqrt_data_matriz,breaks=100)
```



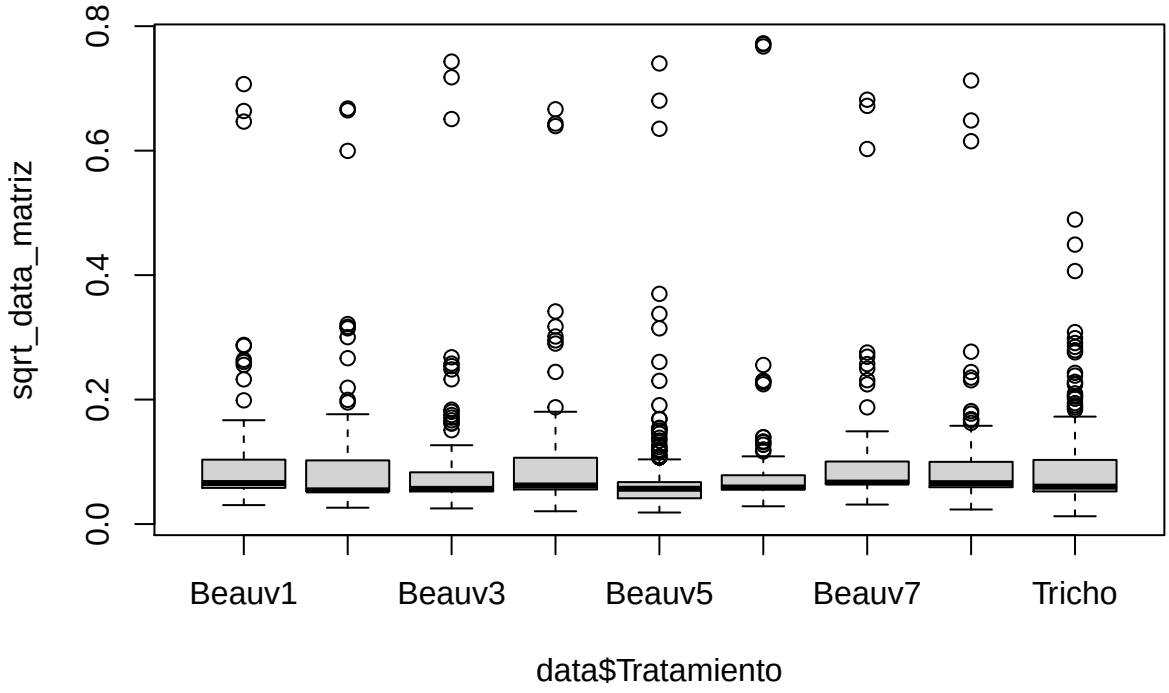

```r
#EJERCICIO: Imputar los ceros con la media de la intensidad de cada tratamiento
media_repeticiones<-apply(data_matriz,1,mean)

for(i in 1:length(media_repeticiones)){
    data_matriz[i,][data_matriz[i,]==0]<-media_repeticiones[i]
}

sqrt_data_matriz<-sqrt(as.matrix(data_matriz))
hist(sqrt_data_matriz,breaks=100)
```



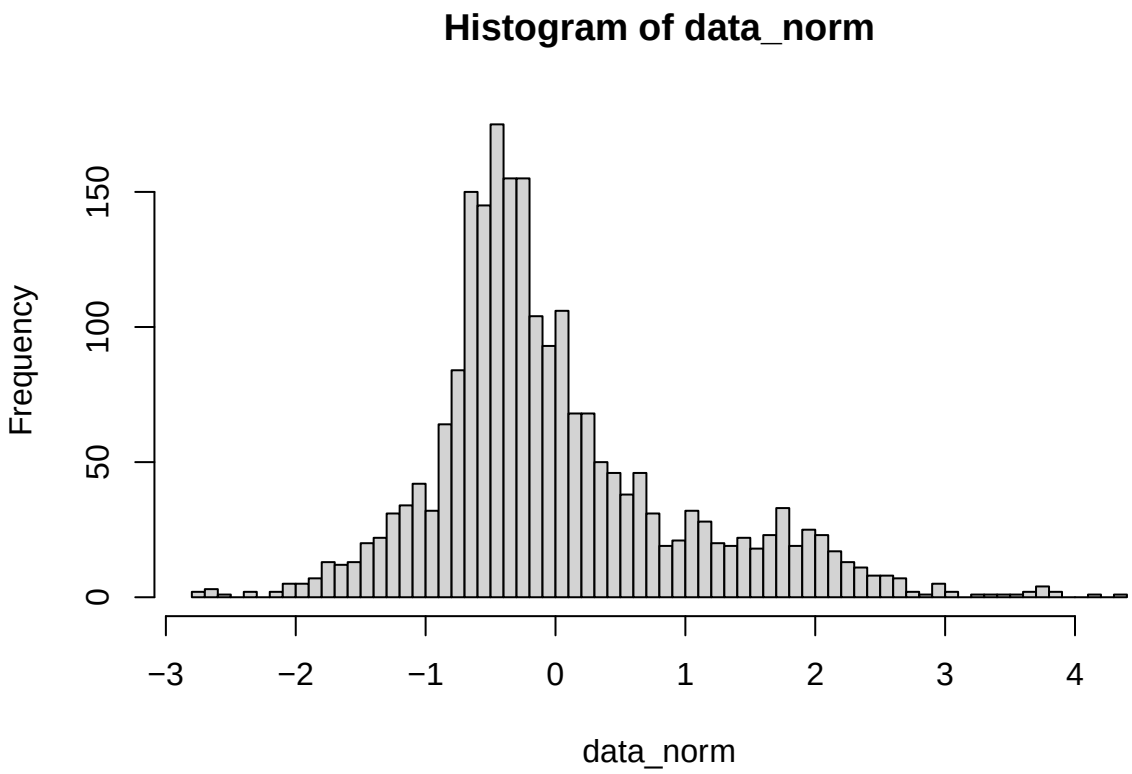

```
#Boxplot intensidades
boxplot(sqrt_data_matriz)
```




```
boxplot(sqrt_data_matriz~data$Tratamiento)
```



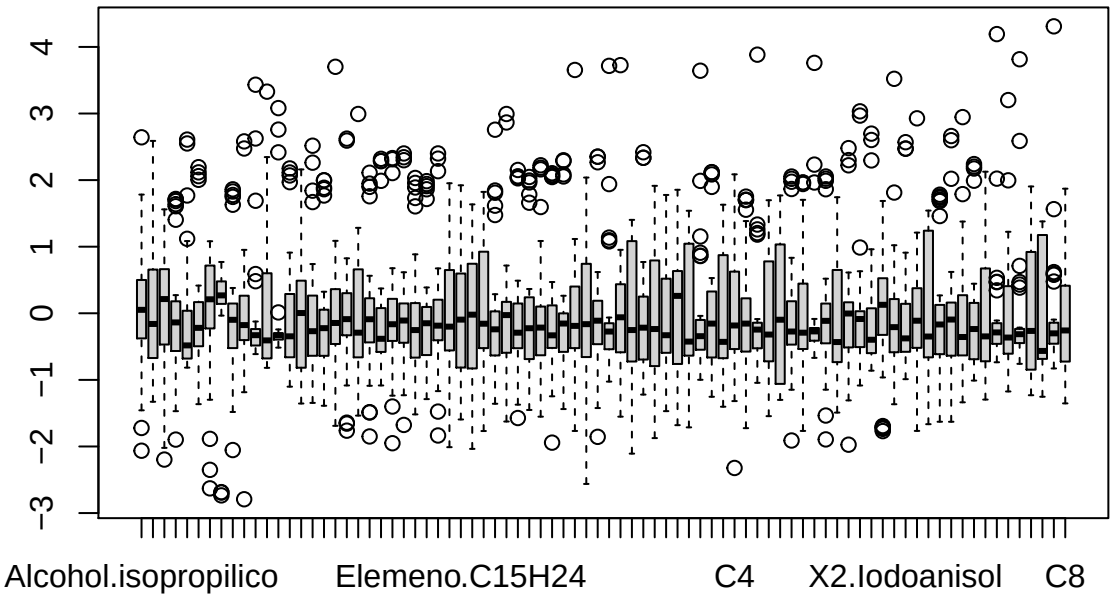

```r
# Normalizar las variables
data_norm <- scale(sqrt_data_matriz,center = T,scale =T)
hist(data_norm,breaks = 100)
```



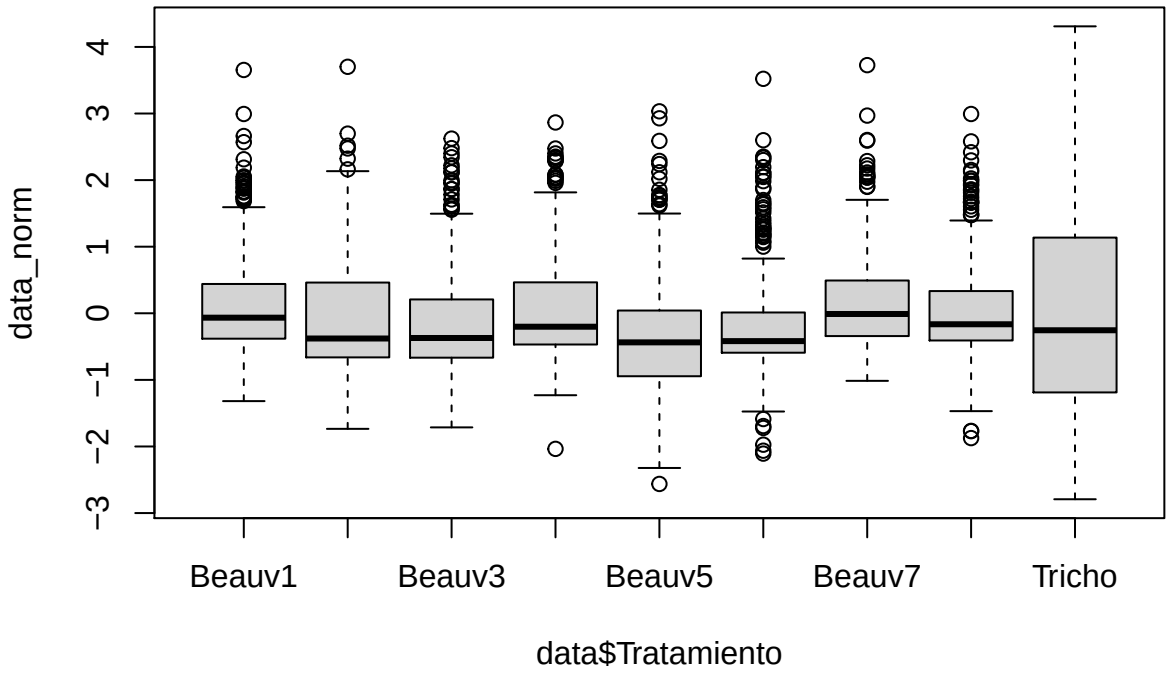

```
dim(data_norm)
```

```
## [1] 27 82
```

```
boxplot(data_norm)
```




```
boxplot(data_norm~data$Tratamiento)
```

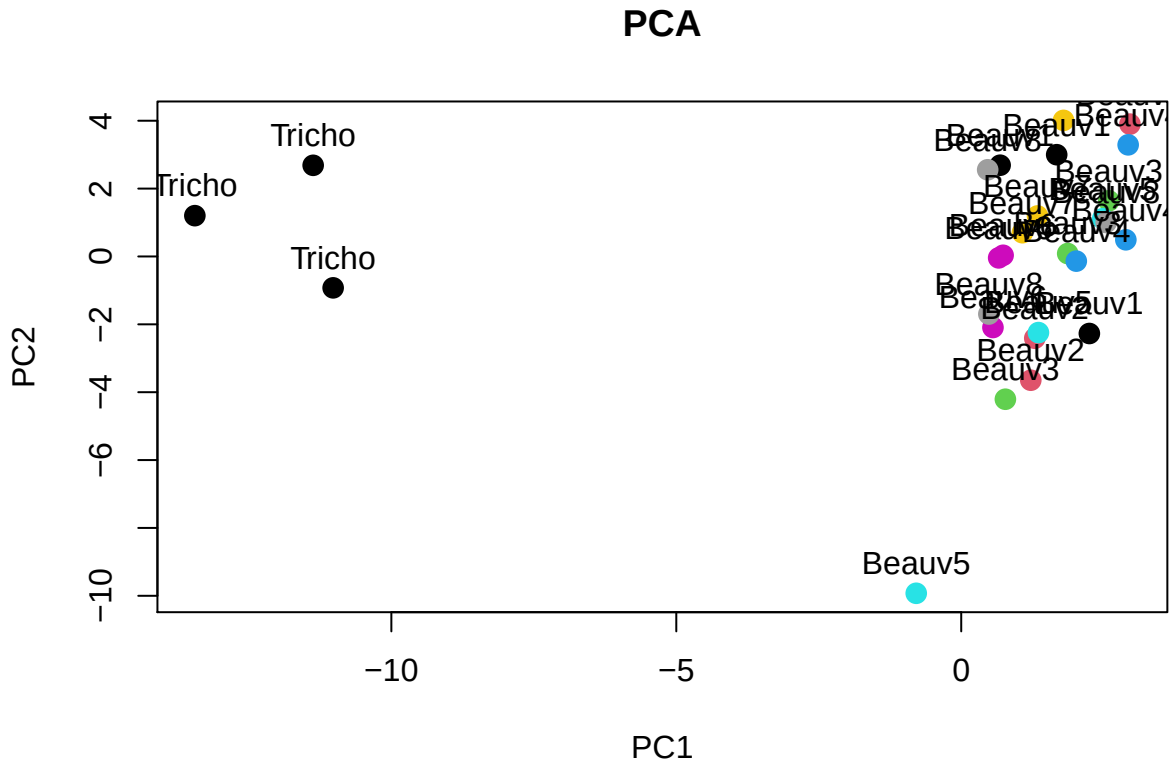



```
##ANÁLISIS DE COMPONENTES PRINCIPALES (PCA)##

# Realizar el análisis de componentes principales
pca <- prcomp(data_norm)
summary(pca)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6     PC7
## Standard deviation     4.4145 3.0018 2.51584 2.13963 2.03217 1.91756 1.85023
## Proportion of Variance 0.2377 0.1099 0.07719 0.05583 0.05036 0.04484 0.04175
## Cumulative Proportion  0.2377 0.3476 0.42474 0.48056 0.53093 0.57577 0.61752
##                           PC8     PC9    PC10    PC11    PC12    PC13   PC14
## Standard deviation     1.79986 1.74797 1.68378 1.64550 1.57608 1.45585 1.4288
## Proportion of Variance 0.03951 0.03726 0.03457 0.03302 0.03029 0.02585 0.0249
## Cumulative Proportion  0.65702 0.69428 0.72886 0.76188 0.79217 0.81802 0.8429
##                           PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     1.33736 1.29550 1.23185 1.19322 1.11697 1.05488 1.01073
## Proportion of Variance 0.02181 0.02047 0.01851 0.01736 0.01521 0.01357 0.01246
## Cumulative Proportion  0.86473 0.88519 0.90370 0.92106 0.93628 0.94985 0.96231
##                           PC22    PC23    PC24    PC25    PC26     PC27
## Standard deviation     0.94853 0.88015 0.81205 0.66833 0.55724 1.96e-15
## Proportion of Variance 0.01097 0.00945 0.00804 0.00545 0.00379 0.00e+00
## Cumulative Proportion  0.97328 0.98272 0.99077 0.99621 1.00000 1.00e+00
```

```
# Graficar las coordenadas de las observaciones en las dos primeras componentes principales
colores<-as.factor(data$Tratamiento)
plot(pca$x[,1], pca$x[,2],
```

```
     main="PCA", xlab="PC1", ylab="PC2",col=as.numeric(colores),
     pch=16,cex=1.5)

# Graficar texto correspondiente
text(pca$x[,1], pca$x[,2],data$Tratamiento,pos=3)
```




```
#Tres componentes principales
library(rgl)
plot3d(x=pca$x[,1],y=pca$x[,2],z=pca$x[,3],
       col=as.numeric(colores),type="s",size=2)

text3d(x=pca$x[,1],y=pca$x[,2],z=pca$x[,3],
       data$Tratamiento,add=T,pos=3)

###### NMDS (Non metric Multidimensional scaling) #######
library(vegan)
```

```
## Loading required package: permute
```

```
## Loading required package: lattice
```

```
## This is vegan 2.5-7
```

```
nmds_result<-metaMDS(data_matriz,distance = "bray",k = 3)
```

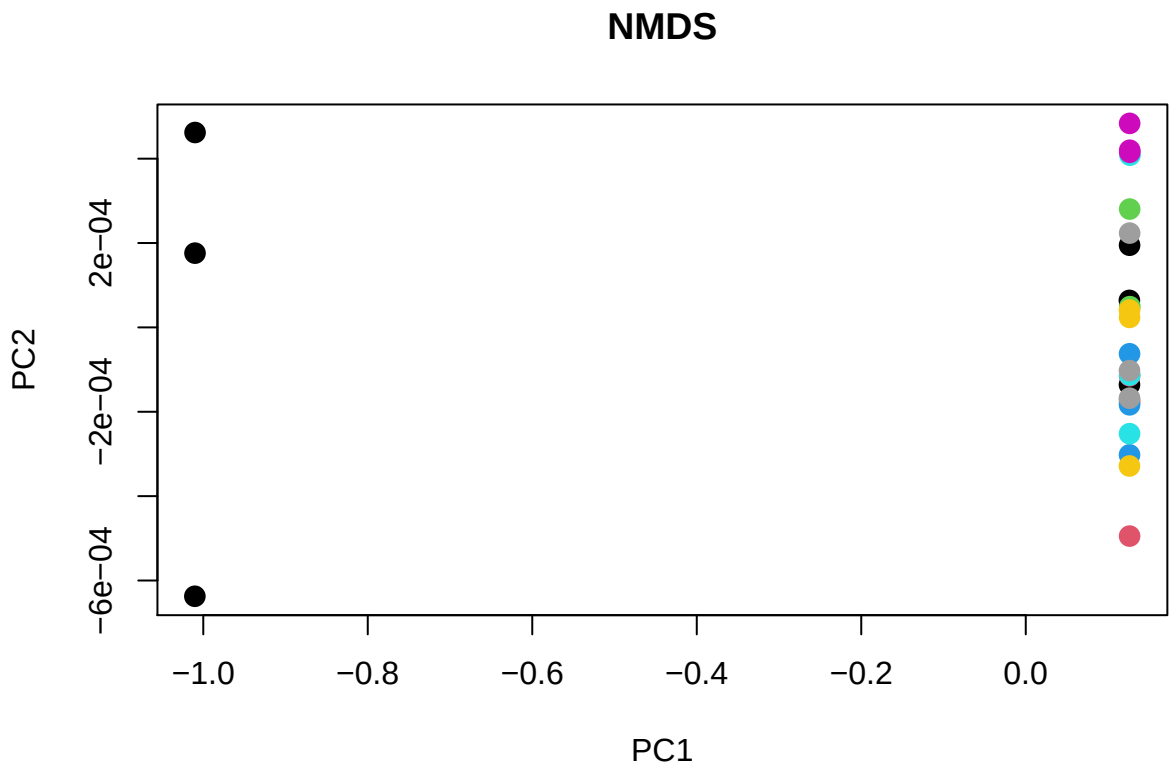
```
## Run 0 stress 9.860019e-05
## Run 1 stress 9.893066e-05
## ... Procrustes: rmse 0.0001432536  max resid 0.0005551023
```

```
## ... Similar to previous best
## Run 2 stress 0.0001120446
## ... Procrustes: rmse 0.000199947  max resid 0.000410323
## ... Similar to previous best
## Run 3 stress 9.700459e-05
## ... New best solution
## ... Procrustes: rmse 4.999051e-05  max resid 0.0001539752
## ... Similar to previous best
## Run 4 stress 9.571237e-05
## ... New best solution
## ... Procrustes: rmse 0.0001796239  max resid 0.0007504895
## ... Similar to previous best
## Run 5 stress 9.90417e-05
## ... Procrustes: rmse 0.0001541354  max resid 0.0003894779
## ... Similar to previous best
## Run 6 stress 9.732141e-05
## ... Procrustes: rmse 0.0001477513  max resid 0.0005246193
## ... Similar to previous best
## Run 7 stress 0.0001560831
## ... Procrustes: rmse 0.000247186  max resid 0.0008683927
## ... Similar to previous best
## Run 8 stress 9.666891e-05
## ... Procrustes: rmse 0.0001104828  max resid 0.0003499816
## ... Similar to previous best
## Run 9 stress 9.957044e-05
## ... Procrustes: rmse 0.0002207158  max resid 0.0007050187
## ... Similar to previous best
## Run 10 stress 9.479829e-05
## ... New best solution
## ... Procrustes: rmse 0.0001439109  max resid 0.0005321248
## ... Similar to previous best
## Run 11 stress 9.864463e-05
## ... Procrustes: rmse 0.0001765452  max resid 0.0007335889
## ... Similar to previous best
## Run 12 stress 9.539153e-05
## ... Procrustes: rmse 0.0001673116  max resid 0.0006795011
## ... Similar to previous best
## Run 13 stress 9.972548e-05
## ... Procrustes: rmse 8.203261e-05  max resid 0.000255562
## ... Similar to previous best
## Run 14 stress 0.0001533494
## ... Procrustes: rmse 0.0002606793  max resid 0.0008360844
## ... Similar to previous best
## Run 15 stress 9.617184e-05
## ... Procrustes: rmse 0.0002028718  max resid 0.0007333457
## ... Similar to previous best
## Run 16 stress 0.0001994973
## ... Procrustes: rmse 0.0002053467  max resid 0.0003904171
## ... Similar to previous best
## Run 17 stress 0.0002361663
## ... Procrustes: rmse 0.0004384847  max resid 0.001279711
## ... Similar to previous best
## Run 18 stress 9.753001e-05
## ... Procrustes: rmse 0.0001814488  max resid 0.00062091
```

```
## ... Similar to previous best
## Run 19 stress 0.0001004289
## ... Procrustes: rmse 0.0001700634  max resid 0.0006611356
## ... Similar to previous best
## Run 20 stress 9.802896e-05
## ... Procrustes: rmse 0.0001803047  max resid 0.0007584757
## ... Similar to previous best
## *** Solution reached
```

```
## Warning in metaMDS(data_matriz, distance = "bray", k = 3): stress is (nearly)
## zero: you may have insufficient data
```

```
plot(nmds_result$points[,1], nmds_result$points[,2],
     main="NMDS", xlab="PC1", ylab="PC2",col=as.numeric(colores),
     pch=16,cex=1.5)
```




```
plot3d(nmds_result$points[,1], nmds_result$points[,2],z=nmds_result$points[,3],
       col=as.numeric(colores),type="s",size=2)

###### PERMANOVA ####################
factor_tratamiento<-as.factor(data$Tratamiento)
adonis(data_matriz~factor_tratamiento,dist="euclidean")
```

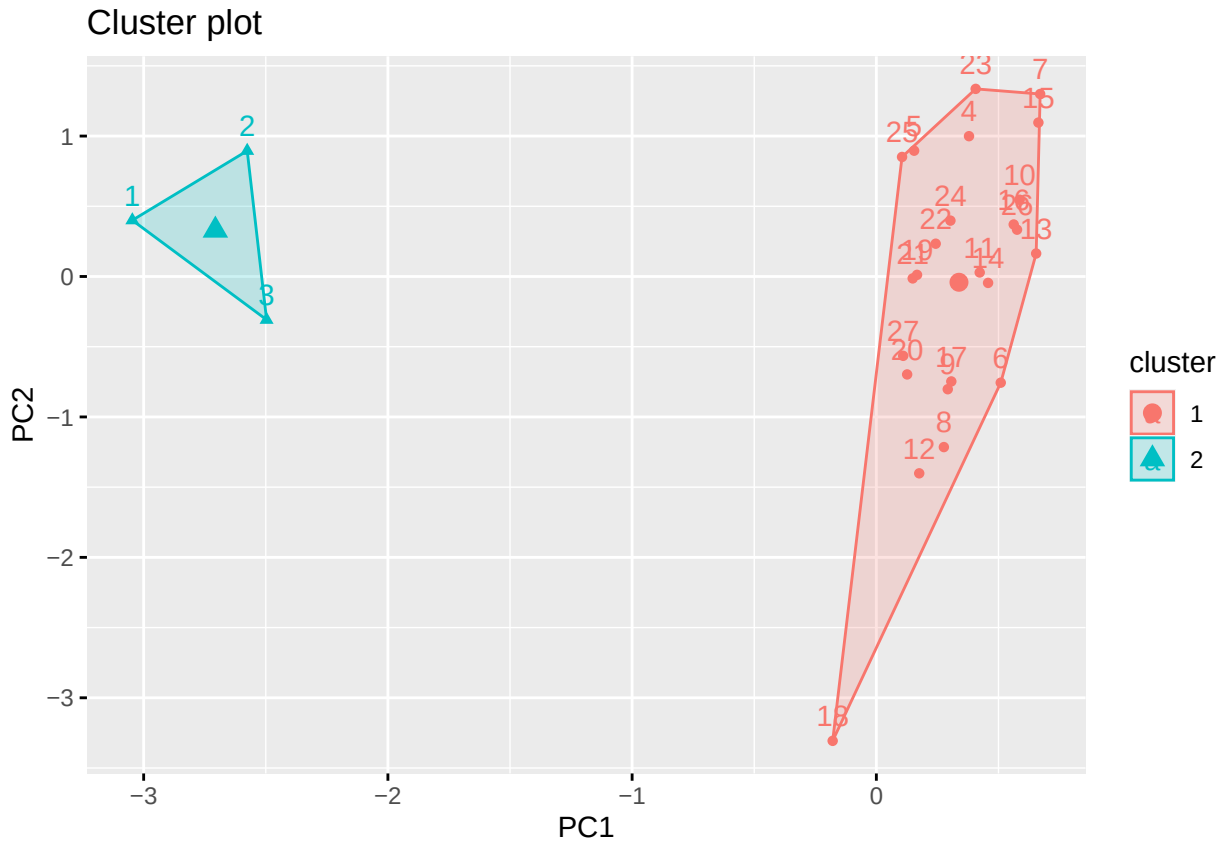
```
##
## Call:
## adonis(formula = data_matriz ~ factor_tratamiento, dist = "euclidean")
##
## Permutation: free
```

```
## Number of permutations: 999
##
## Terms added sequentially (first to last)
##
##                     Df SumsOfSqs  MeanSqs F.Model      R2 Pr(>F)
## factor_tratamiento  8   1.23579 0.154473  8.5047 0.79079  0.001 ***
## Residuals          18   0.32694 0.018163         0.20921
## Total              26   1.56273                  1.00000
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

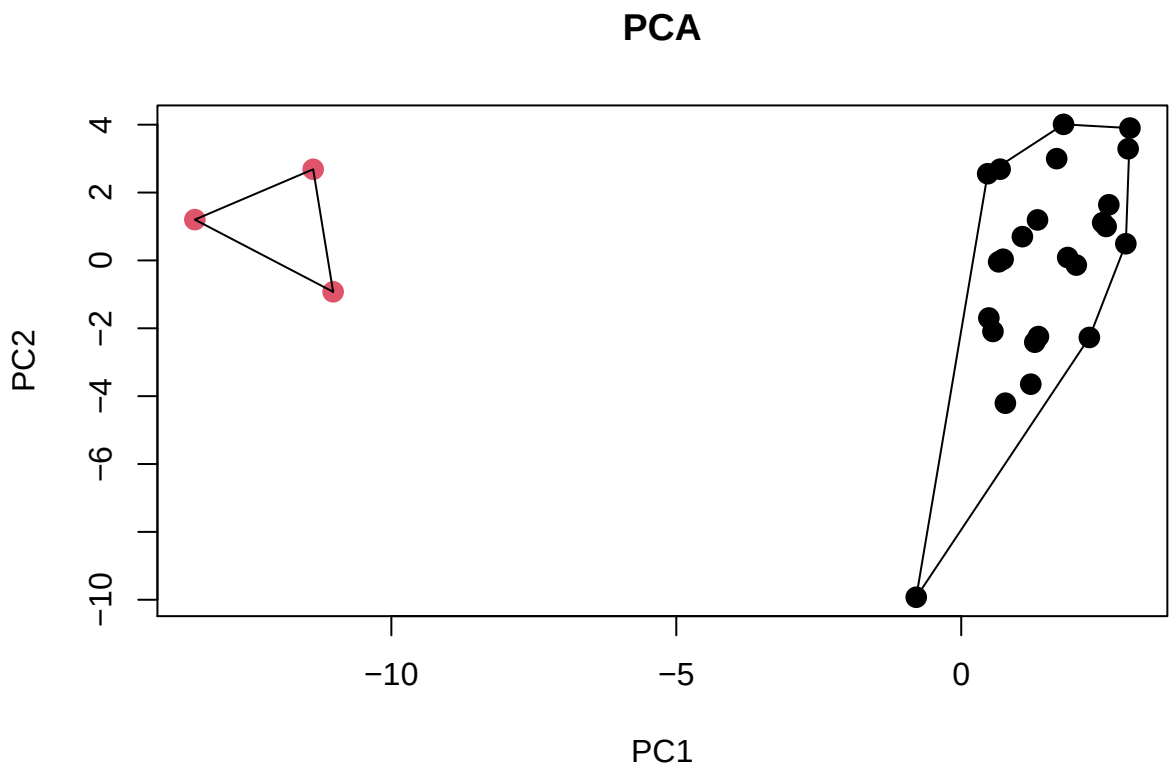
```
###### k-medias ########################
library(ggplot2)
library(factoextra)
```

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

```
coordenadas<-data.frame(pca$x[,c(1,2)])
kmedias<-kmeans(coordenadas,centers = 2,nstart = 25)
fviz_cluster(kmedias,data=coordenadas)
```




```
plot(pca$x[,1], pca$x[,2],
     main="PCA", xlab="PC1", ylab="PC2", col=kmedias$cluster,
     pch=16,cex=1.5)
ordihull(pca,groups = kmedias$cluster)
```




```
##### En búsqueda de biomarcadores ########
loadings_contribucion<-pca$rotation[,c(1,2)]
write.csv(loadings_contribucion,"loadings.csv")
##Podemos hacerlo manualmente...o crearnos una función para automatizar la tarea

##LLAMAR A LA FUNCIÓN loadbiomarc()
loadbiomarc<-function(loadings_contribucion1,data_norm1,n_biom = 10){ #Parámetros

  nombres = vector() #Creamos un vector vacio llamado nombres
  for(i in 1:ncol(loadings_contribucion1)){ #Para cada i (columna) en loadings_contribucion
    contribucion = loadings_contribucion1[,c(i)] #Extraer la i-esima columna y ponerla en el vector con
    orden = order(abs(contribucion),decreasing = T) #ordenar los valores absolutos de cada contribucion
    nombres = c(nombres,names(contribucion)[head(orden,n = n_biom)]) #guardar los nombres ordenados en
  }
  #despues...
  if(any(duplicated(nombres)) == TRUE){ #Si alguno de los nombres está duplicado
    nombres = nombres[!duplicated(nombres)] #conservar solo uno de los nombres y crear un vector sin no
  } else { #y si no hay duplicados...
    nombres = nombres #usar el mismo vector de nombres
  }
  #despues...
  matriz_simplificada = data_norm1[,nombres] #extraer las columnas cuyos nombres coincidan  con los nom
  return(matriz_simplificada) #...y finalmente regresar la matriz de las intensidades correspondientes
}

data_biomar<-loadbiomarc(loadings_contribucion,data_norm,n_biom = 20)
```

```r
dim(data_biomar)
```

```
## [1] 27 37
```

```r
colnames(data_biomar)
```

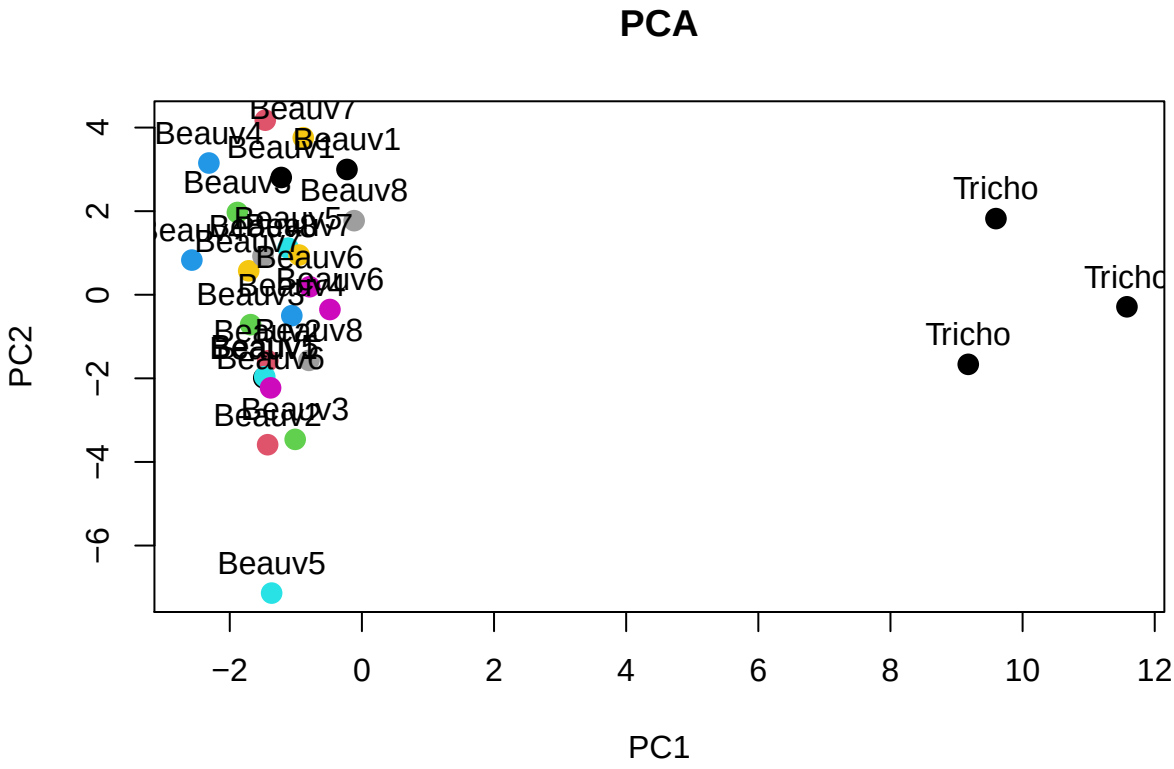
```
##  [1] "X1.Butanol..3.methyl...formate"
##  [2] "X3.Metil.butanol"
##  [3] "Cyclohexene..3..1.5.dimethyl.4.hexenyl..6.methylene....S..R.S...beta.Sesquiphellandrene.1"
##  [4] ".beta..Phellandrene"
##  [5] "X2H.Pyran.2.one..6.pentyl."
##  [6] "X1.Propanol"
##  [7] "Epizonarene.1"
##  [8] "X1.3.Cyclohexadiene..5..1.5.dimethyl.4.hexenyl..2.methyl....S..R.S.......Zingiberene"
##  [9] "Acetoina"
## [10] "Bicyclo.9.3.1.pentadeca.3.7.dien.12.ol..4.8.12.15.15.pentamethyl....1R..1.R.3E.7E.11R.12R....V
## [11] "X2.Metil.propanol"
## [12] "X2.Cyclohexen.1.one..3.methyl.6..1.methylethenyl.....S..p.Mentha.1.8.dien.3.one"
## [13] "Bicyclo.3.1.0.hex.2.ene..4.methyl.1..1.methylethyl....beta..Phellandrene"
## [14] "X2.5.Dimetilfurano"
## [15] "X1.3.Cyclohexadiene..5..1.5.dimethyl.4.hexenyl..2.methyl....S..R.S.......Zingiberene.1"
## [16] "Cyclohexene..3..1.5.dimethyl.4.hexenyl..6.methylene....S..R.S...beta.Sesquiphellandrene"
## [17] "Methoxyacetic.acid..tetradecyl.ester"
## [18] "X1.4.Cyclohexadiene..1.methyl.4..1.methylethyl...gamma.terpineno"
## [19] "Camphene"
## [20] "X2.4.Hexadiene"
## [21] "X1H.3a.7.Methanoazulene..2.3.4.7.8.8a.hexahydro.3.6.8.8.tetramethyl....3R..3.alpha..3a.beta..7
## [22] "X3.Furanmethanol"
## [23] "X1H.Benzocycloheptene..2.4a.5.6.7.8.hexahydro.3.5.5.9.tetramethyl....R...beta.Himachalene"
## [24] "Epizonarene"
## [25] "gamma.Gurjeneno"
## [26] "Butanoic.acid..3.hydroxy...ethyl.ester"
## [27] "X1.6.10.Dodecatrien.3.ol..3.7.11.trimethyl....E.."
## [28] "Etanol"
## [29] "X2.Isopropenil.4a.8.dimetil.1.2.3.4.4a.5.6.7.octahidronaftaleno"
## [30] "X1.3.Cyclohexadiene..1.methyl.4..1.methylethyl..alpha.terpineno"
## [31] "X2.Undecanol"
## [32] "beta.Selineno"
## [33] "Thujopsene.I3"
## [34] "C8"
## [35] "X1H.3a.7.Methanoazulene..2.3.4.7.8.8a.hexahydro.3.6.8.8.tetramethyl....3R..3.alpha..3a.beta..7
## [36] "n.Decilsulfona"
## [37] "X3.3.Iminobispropylamine"
```

```r
#### Ahora reconstruimos el PCA unicamente con los biomarcadores
pca2 <- prcomp(data_biomar)
summary(pca2)
```

```
## Importance of components:
##                           PC1    PC2     PC3     PC4     PC5     PC6    PC7
## Standard deviation     3.7026 2.5569 1.71630 1.48802 1.30620 1.17820 1.0745
## Proportion of Variance 0.3705 0.1767 0.07961 0.05984 0.04611 0.03752 0.0312
## Cumulative Proportion  0.3705 0.5472 0.62684 0.68668 0.73280 0.77031 0.8015
##                           PC8     PC9    PC10   PC11    PC12   PC13    PC14
## Standard deviation     1.04308 0.99233 0.93882 0.9246 0.83465 0.8160 0.65388
```

```
## Proportion of Variance 0.02941 0.02661 0.02382 0.0231 0.01883 0.0180 0.01156
## Cumulative Proportion  0.83092 0.85754 0.88136 0.9045 0.92329 0.9413 0.95284
##                           PC15    PC16    PC17    PC18    PC19    PC20    PC21
## Standard deviation     0.61305 0.53088 0.51572 0.45344 0.37844 0.37533 0.32813
## Proportion of Variance 0.01016 0.00762 0.00719 0.00556 0.00387 0.00381 0.00291
## Cumulative Proportion  0.96300 0.97062 0.97781 0.98336 0.98723 0.99104 0.99395
##                           PC22   PC23    PC24    PC25    PC26      PC27
## Standard deviation     0.28951 0.2433 0.21658 0.16595 0.07945 3.395e-16
## Proportion of Variance 0.00227 0.0016 0.00127 0.00074 0.00017 0.000e+00
## Cumulative Proportion  0.99622 0.9978 0.99909 0.99983 1.00000 1.000e+00
```

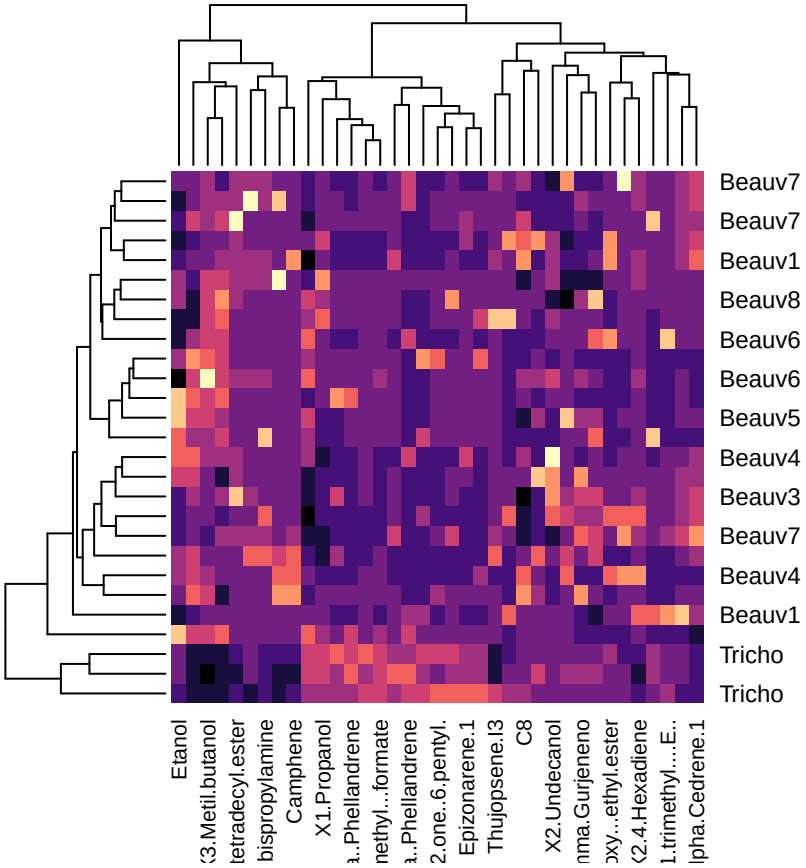
```
colores<-as.factor(data$Tratamiento)
plot(pca2$x[,1], pca2$x[,2],
     main="PCA", xlab="PC1", ylab="PC2",col=as.numeric(colores),
     pch=16,cex=1.5)
text(pca2$x[,1], pca2$x[,2],data$Tratamiento,pos=3)
```




```
plot3d(x=pca2$x[,1],y=pca2$x[,2],z=pca2$x[,3],
       col=as.numeric(colores),type="s",size=2)

text3d(x=pca2$x[,1],y=pca2$x[,2],z=pca2$x[,3],
       data$Tratamiento,add=T,pos=3)

#### Mapa de calor de compuestos biomarcadores #####
class(data_biomar)
```

```
## [1] "matrix" "array"
```

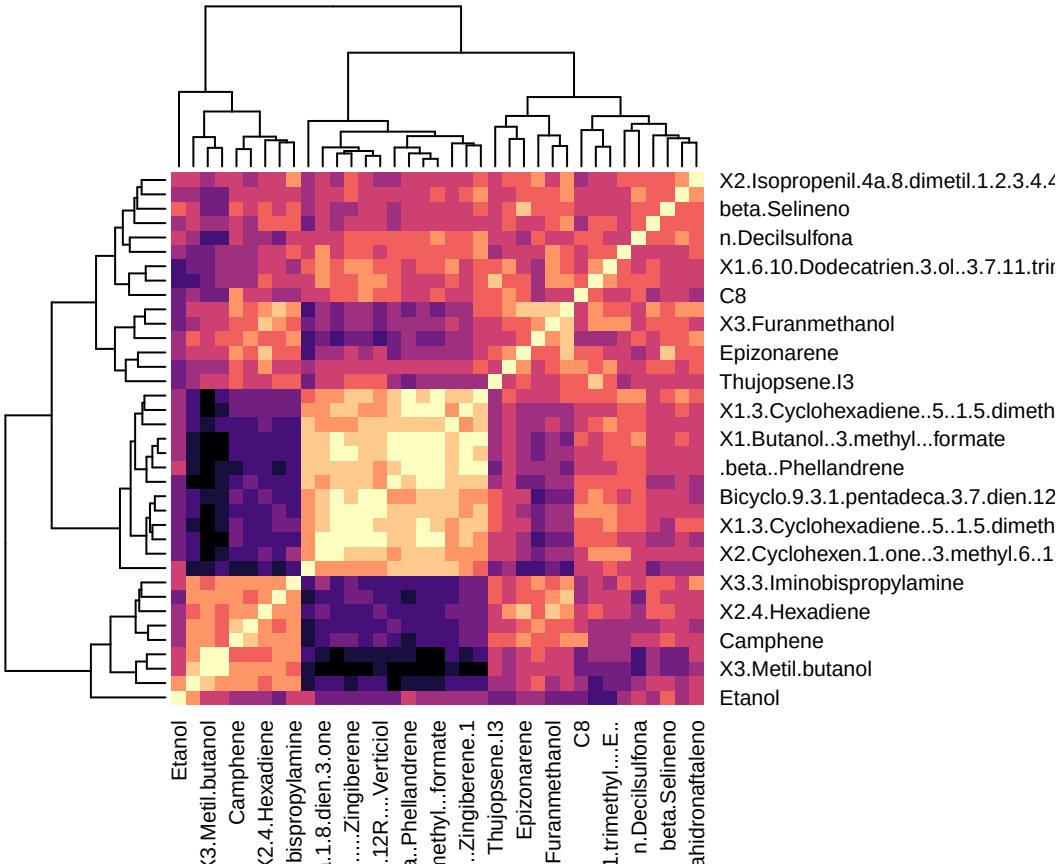
```
row.names(data_biomar) <- colores
dist_renglones<-vegdist(data_biomar,method = "euclidean")
dist_columnas<-vegdist(t(data_biomar),method = "euclidean")
heatmap(x = data_biomar,
        Rowv = dist_renglones,
        Colv = dist_columnas,
        col=magma(10))
```




```
#ver: https://cran.r-project.org/web/packages/viridis/vignettes/intro-to-viridis.html

#### Correlogramas ####
corr_matriz<-cor(data_biomar)
dim(corr_matriz)
```

```
## [1] 37 37
```

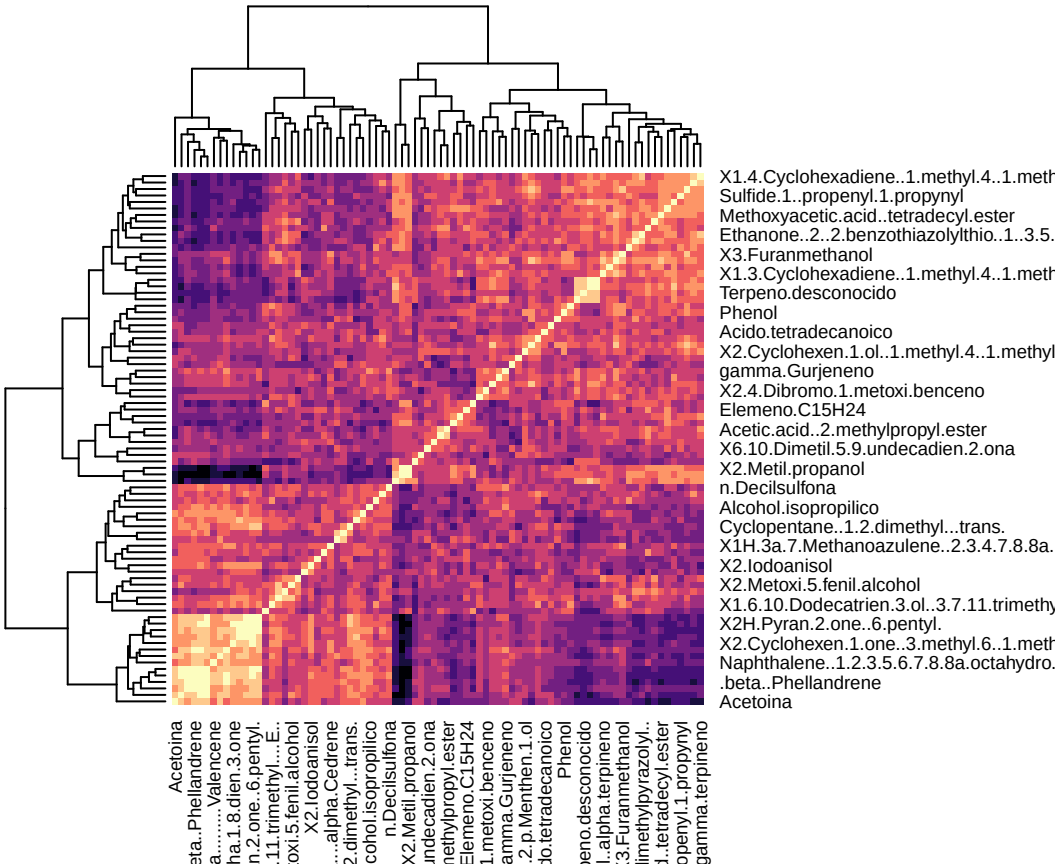
```
dist<-vegdist(corr_matriz,method = "euclidean")
heatmap(x = corr_matriz,
        Rowv = dist,
        Colv = dist,
        col=magma(10),
        symm = T)
```




```
corr_matriz<-cor(data_norm)
dim(corr_matriz)
```

```
## [1] 82 82
```

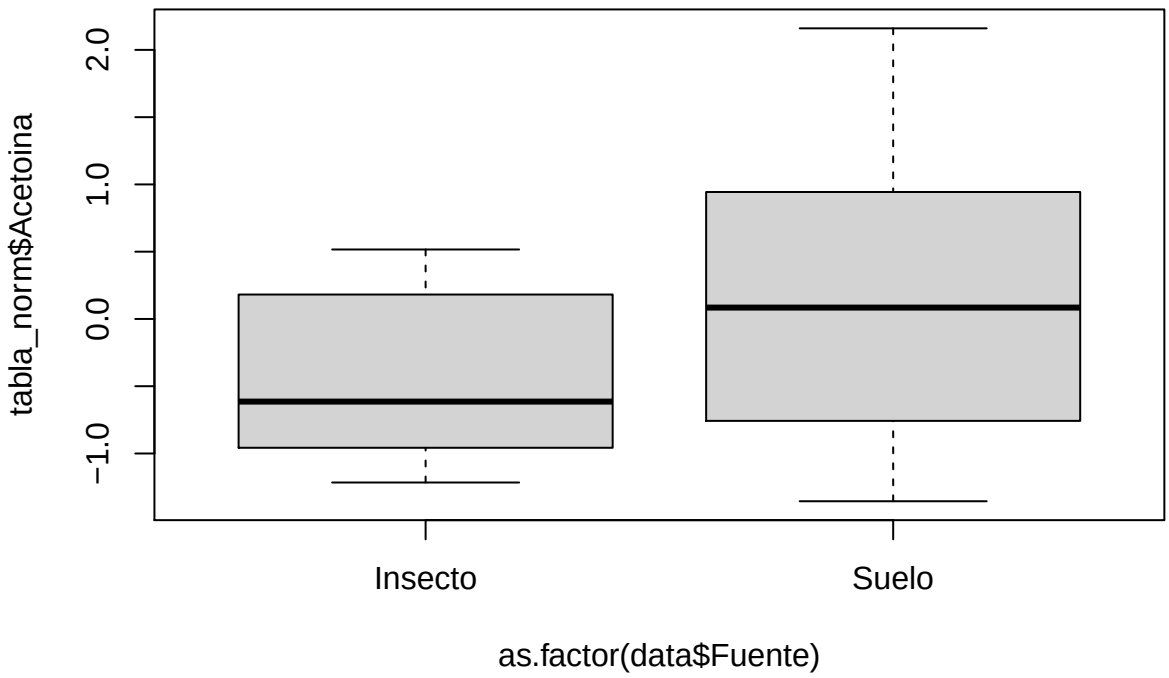
```
dist<-vegdist(corr_matriz,method = "euclidean")
heatmap(x = corr_matriz,
        Rowv = dist,
        Colv = dist,
        col=magma(10),
        symm = T)
```




```
### Bosques aleatorios (random forest) ###
library(randomForest)
```

```
## randomForest 4.6-14
## Type rfNews() to see new features/changes/bug fixes.
##
## Attaching package: 'randomForest'
## The following object is masked from 'package:ggplot2':
##
##     margin
```

```
modelo_rf <- randomForest(as.factor(data$Fuente) ~ .,
                          data = data_norm, ntree = 1000,keep.forest=F,importance=T)
contribucion <- importance(modelo_rf)
write.csv(contribucion,"random_forest.csv")

########## BOXPLOT ALGUNOS METABOLITOS ######################
tabla_norm<-as.data.frame(data_norm)
boxplot(tabla_norm$Acetoina~as.factor(data$Fuente))
```




```
##### FIN INTRODUCCION #####
```