Módulo 5 Procesamiento de Datos Multivariados

Franco Quintanilla

2022-10-18

Instalamos las librerías que vamos a utilizar

```
library(MVN)
library(ggplot2)
library(ggcorrplot)
library(stats)
library(factoextra)
```

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

```
library(FactoMineR)
```

Importamos los datos

```
df = read.csv("/Users/francoquintanilla/Documents/R/mercurio.csv", row.names=1)
head(df)
```

```
##             X2    X3  X4   X5    X6   X7 X8   X9  X10  X11 X12
## 1    Alligator   5.9 6.1  3.0   0.7 1.23  5 0.85 1.43 1.53   1
## 2        Annie   3.5 5.1  1.9   3.2 1.33  7 0.92 1.90 1.33   0
## 3       Apopka 116.0 9.1 44.1 128.3 0.04  6 0.04 0.06 0.04   0
## 4 Blue Cypress  39.4 6.9 16.4   3.5 0.44 12 0.13 0.84 0.44   0
## 5        Brick   2.5 4.6  2.9   1.8 1.20 12 0.69 1.50 1.33   1
## 6       Bryant  19.6 7.3  4.5  44.1 0.27 14 0.04 0.48 0.25   1
```

Limpieza de datos

Primero, vamos a crear una función, la cual nos va a limpiar nuestro dataset en base a los cuantiles.

```
f_outliers = function(x, removeNA = TRUE)
  {
  qrts = quantile(x, probs=c(0.25, 0.75), na.rm=removeNA)
  caps = quantile(x, probs=c(0.05, 0.95), na.rm=removeNA)
  iqr = qrts[2] - qrts[1]
  x[x<qrts[1] - 1.5*iqr] = caps[1]
  x[x>qrts[2] + 1.5*iqr] = caps[2]
  x
  }
```

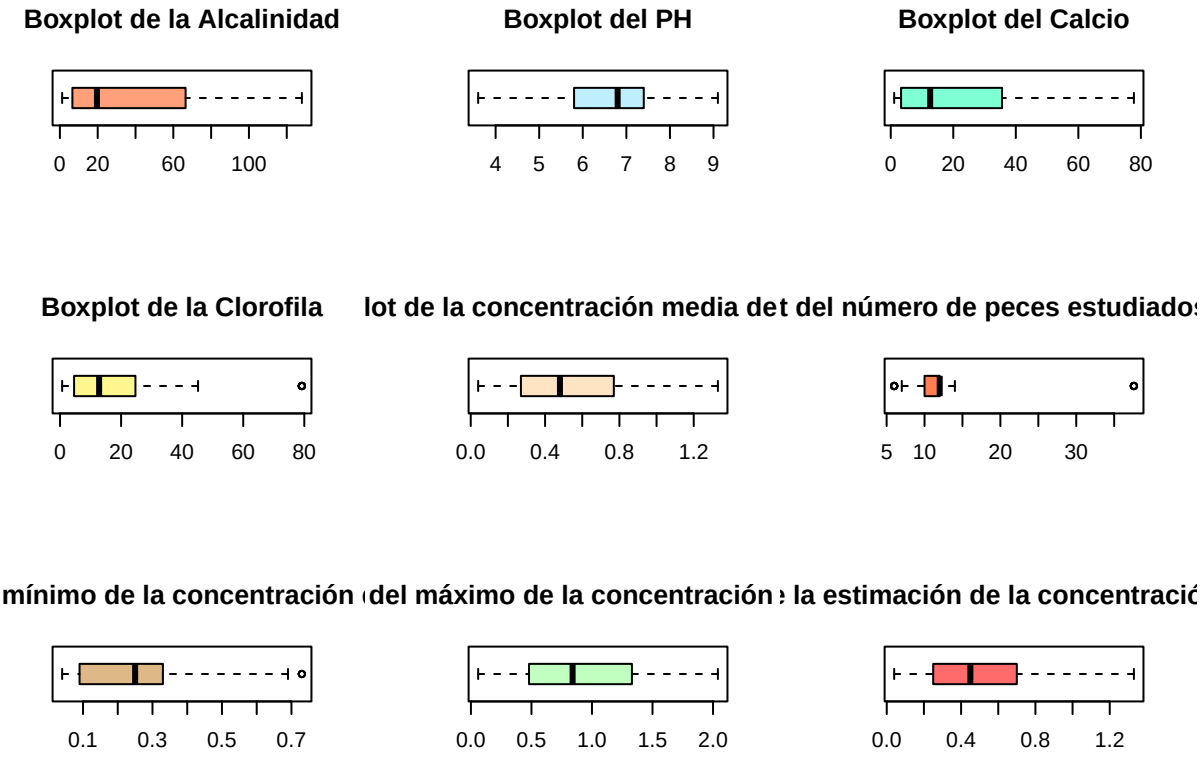
Ahora, vamos a pasar los datos por está función para eliminar los outliers para que no nos hagan ruido estos valores.

```
x3 = f_outliers(df$X3)
x4 = f_outliers(df$X4)
x5 = f_outliers(df$X5)
```

```
x6 = f_outliers(df$X6)
x7 = f_outliers(df$X7)
x8 = f_outliers(df$X8)
x9 = f_outliers(df$X9)
x10 = f_outliers(df$X10)
x11 = f_outliers(df$X11)
```

Vamos a corroborar la limpieza de los datos y visualizarlos con boxplots.

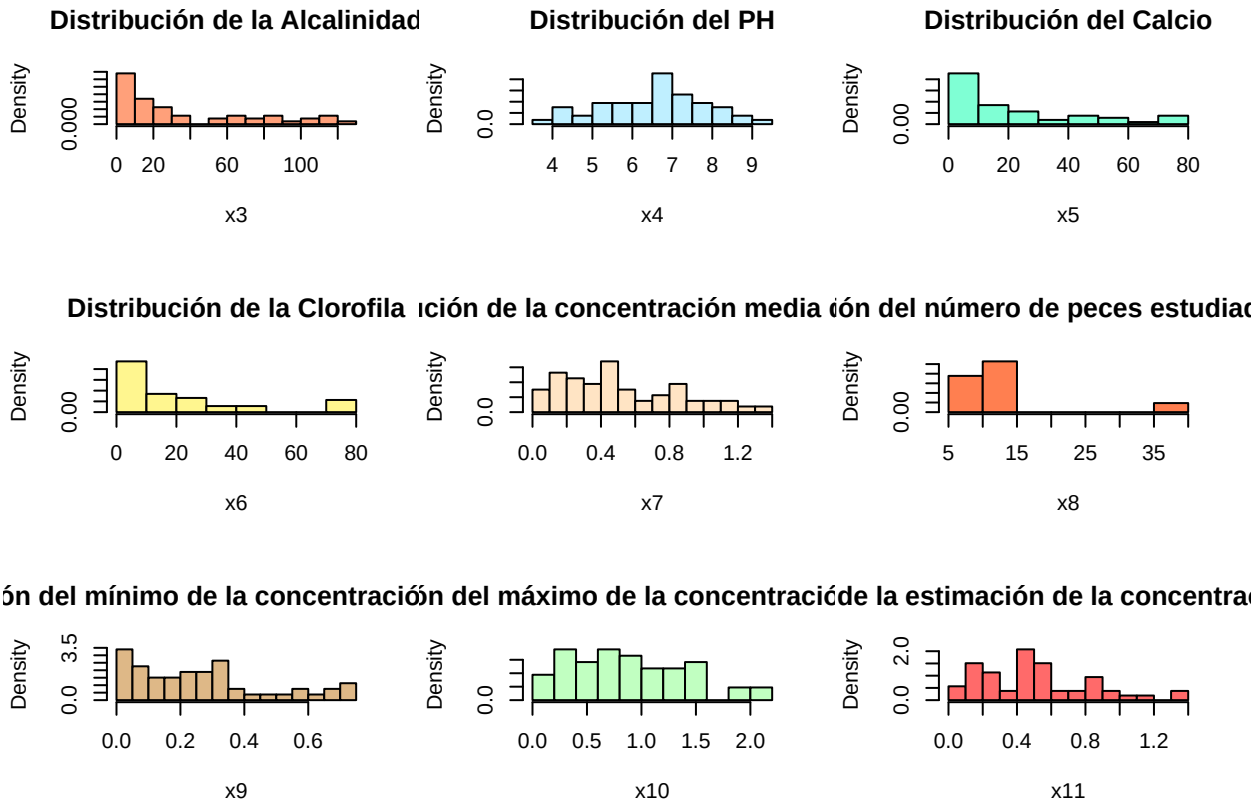
```
par(mfrow=c(3,3))
boxplot(x3, col="#FFA07A", main="Boxplot de la Alcalinidad",
        horizontal=TRUE)
boxplot(x4, col="#BFEFFF", main="Boxplot del PH",
        horizontal=TRUE)
boxplot(x5, col="#7FFFD4", main="Boxplot del Calcio",
        horizontal=TRUE)
boxplot(x6, col="#FFF68F", main="Boxplot de la Clorofila",
        horizontal=TRUE)
boxplot(x7, col="#FFE4C4", main="Boxplot de la concentración media de mercurio",
        horizontal=TRUE)
boxplot(x8, col="#FF7F50", main="Boxplot del número de peces estudiados en el lago",
        horizontal=TRUE)
boxplot(x9, col="#DEB887", main="Boxplot del mínimo de la concentración de mercurio",
        horizontal=TRUE)
boxplot(x10, col="#C1FFC1", main="Boxplot del máximo de la concentración de mercurio",
        horizontal=TRUE)
boxplot(x11, col="#FF6A6A", main="Boxplot de la estimación de la concentración de mercurio",
        horizontal=TRUE)
```



Análisis de Normalidad

Ahora, para el análisis de normalidad, primero vamos a visualizar su comportamiento con histogramas.

```
par(mfrow=c(3,3))
hist(x3, col="#FFA07A", main="Distribución de la Alcalinidad",
     breaks=10, freq=FALSE)
hist(x4, col="#BFEFFF", main="Distribución del PH",
     breaks=10, freq=FALSE)
hist(x5, col="#7FFFD4", main="Distribución del Calcio",
     breaks=10, freq=FALSE)
hist(x6, col="#FFF68F", main="Distribución de la Clorofila",
     breaks=10, freq=FALSE)
hist(x7, col="#FFE4C4", main="Distribución de la concentración media de mercurio",
     breaks=10, freq=FALSE)
hist(x8, col="#FF7F50", main="Distribución del número de peces estudiados en el lago",
     breaks=10, freq=FALSE)
hist(x9, col="#DEB887", main="Distribución del mínimo de la concentración de mercurio",
     breaks=10, freq=FALSE)
hist(x10, col="#C1FFC1", main="Distribución del máximo de la concentración de mercurio",
     breaks=10, freq=FALSE)
hist(x11, col="#FF6A6A", main="Distribución de la estimación de la concentración de mercurio",
     breaks=10, freq=FALSE)
```



Como podemos ver, ninguno de los valores parece tener un comportamiento normal, hay algunos que pueden tener una tendencia como el PH, o el máximo de la concentración de mercurio, pero para eso, tenemos que hacer unas pruebas de normalidad.

Antes que hacer otra cosa, volvemos a crear un dataframe con los valores limpios y los valores que nos interese hacer el análisis.

```r
df2 = data.frame("Alcalinidad"=x3, "PH"=x4, "Calcio"=x5, "Clorofila"=x6,
                 "Min_Conc"=x9, "Max_Conc"=x10, "Est_Conc"=x11)
df2
```

```
##    Alcalinidad  PH Calcio Clorofila Min_Conc Max_Conc Est_Conc
## 1          5.9 6.1   3.00       0.7     0.73     1.43    1.144
## 2          3.5 5.1   1.90       3.2     0.73     1.90    1.330
## 3        116.0 9.1  44.10      79.2     0.04     0.06    0.040
## 4         39.4 6.9  16.40       3.5     0.13     0.84    0.440
## 5          2.5 4.6   2.90       1.8     0.69     1.50    1.330
## 6         19.6 7.3   4.50      44.1     0.04     0.48    0.250
## 7          5.2 5.4   2.80       3.4     0.30     0.72    0.450
## 8         71.4 8.1  55.20      33.7     0.08     0.38    0.160
## 9         26.4 5.8   9.20       1.6     0.26     1.40    0.720
## 10         4.8 6.4   4.60      22.5     0.41     1.47    0.810
## 11         6.6 5.4   2.70      14.9     0.52     0.86    0.710
## 12        16.5 7.2  13.80       4.0     0.10     0.73    0.510
## 13        25.4 7.2  25.20      11.6     0.26     1.01    0.540
## 14         7.1 5.8   5.20       5.8     0.50     2.03    1.000
## 15       128.0 7.6  77.76      79.2     0.04     0.11    0.050
## 16        83.7 8.2  66.50      79.2     0.12     0.18    0.150
## 17       108.5 8.7  35.60      79.2     0.07     0.43    0.190
## 18        61.3 7.8  57.40      13.9     0.32     1.50    0.490
## 19         6.4 5.8   4.00       4.6     0.64     1.33    1.020
## 20        31.0 6.7  15.00      17.0     0.67     1.44    0.700
## 21         7.5 4.4   2.00       9.6     0.33     0.93    0.450
## 22        17.3 6.7  10.70       9.5     0.37     0.94    0.590
## 23        12.6 6.1   3.70      21.0     0.25     0.61    0.410
## 24         7.0 6.9   6.30      32.1     0.33     2.04    0.810
## 25        10.5 5.5   6.30       1.6     0.25     0.62    0.420
## 26        30.0 6.9  13.90      21.5     0.23     1.12    0.530
## 27        55.4 7.3  15.90      24.7     0.17     0.52    0.310
## 28         3.9 4.5   3.30       7.0     0.59     1.38    0.870
## 29         5.5 4.8   1.70      14.8     0.31     0.84    0.500
## 30         6.3 5.8   3.30       0.7     0.19     0.69    0.470
## 31        67.0 7.8  58.60      43.8     0.16     0.59    0.250
## 32        28.8 7.4  10.20      32.7     0.16     0.65    0.410
## 33         5.8 3.6   1.60       3.2     0.31     1.90    0.870
## 34         4.5 4.4   1.10       3.2     0.25     1.02    0.560
## 35       119.1 7.9  38.40      16.1     0.07     0.30    0.160
## 36        25.4 7.1   8.80      45.2     0.09     0.29    0.160
## 37       106.5 6.8  77.76      16.5     0.05     0.37    0.230
## 38        53.0 8.4  45.60      79.2     0.04     0.06    0.040
## 39         8.5 7.0   2.50      12.8     0.31     0.63    0.560
## 40        87.6 7.5  77.76      20.1     0.73     1.41    0.890
## 41       114.0 7.0  72.60       6.4     0.04     0.26    0.180
## 42        97.5 6.8  45.50       6.2     0.05     0.26    0.190
## 43        11.8 5.9  24.20       1.6     0.27     1.05    0.440
## 44        66.5 8.3  26.00      79.2     0.05     0.48    0.160
## 45        16.0 6.7  41.20      24.1     0.36     1.40    0.670
## 46         5.0 6.2  23.60       9.6     0.31     0.95    0.550
## 47        25.6 6.2  12.60      27.7     0.30     1.10    0.580
## 48        81.5 8.9  20.50       9.6     0.04     0.40    0.270
## 49         1.2 4.3   2.10       6.4     0.59     1.24    0.980
```

```
## 50        34.0 7.0  13.10       4.6     0.08     0.90    0.310
## 51        15.5 6.9   5.20      16.5     0.23     0.69    0.430
## 52        17.3 5.2   3.00       2.6     0.15     0.40    0.280
## 53        71.8 7.9  20.50       8.8     0.15     0.51    0.250
```

Una vez que tenemos el nuevo data frame, ahora si podemos hacer las pruebas de normalidad

Prueba de Normalidad de Mardia

```
n_test = mvn(df2, mvnTest="mardia")
n_test$multivariateNormality
```

```
##              Test        Statistic              p value Result
## 1 Mardia Skewness 184.544953319842 1.65571079235445e-09     NO
## 2 Mardia Kurtosis  1.9860226693287   0.0470308068283103     NO
## 3             MVN             <NA>                 <NA>     NO
```

Como podemos observar en este caso, no pasan la prueba de normalidad de Mardia, esto en base a los resultados de la curtosis y el sesgo que presentan.

Prueba de Normalidad de Anderson Darling

```
n_test$univariateNormality
```

```
##               Test    Variable Statistic   p value Normality
## 1 Anderson-Darling Alcalinidad    3.6725  <0.001      NO
## 2 Anderson-Darling     PH         0.3496  0.4611      YES
## 3 Anderson-Darling   Calcio       3.9790  <0.001      NO
## 4 Anderson-Darling  Clorofila     4.7492  <0.001      NO
## 5 Anderson-Darling  Min_Conc      1.8380   1e-04      NO
## 6 Anderson-Darling  Max_Conc      0.6585   0.081      YES
## 7 Anderson-Darling  Est_Conc      0.8640  0.0248      NO
```

En el caso de la prueba de ***Anderson-Darling***, nos dice que nuestras conjeturas fueron correctas, que tanto el *PH* como el *Máximo de la concentración de mercurio* tienen un comportamiento normal, por lo que pasaron el test, las demás variables no tienen ese comportamiento.

También podemos observar los resultados de las demás medidas descriptivas.

```
n_test$Descriptives
```

```
##              n       Mean    Std.Dev Median  Min    Max 25th  75th       Skew
## Alcalinidad 53 37.5301887 38.2035267  19.60 1.20 128.00 6.60 66.50  0.9679170
## PH          53  6.5905660  1.2884493   6.80 3.60   9.10 5.80  7.40 -0.2458771
## Calcio      53 21.6467925 23.5076995  12.60 1.10  77.76 3.30 35.60  1.1519903
## Clorofila   53 21.1641509 23.8639743  12.80 0.70  79.20 4.60 24.70  1.5167803
## Min_Conc    53  0.2728302  0.2091961   0.25 0.04   0.73 0.09  0.33  0.8259845
## Max_Conc    53  0.8745283  0.5220469   0.84 0.06   2.04 0.48  1.33  0.4645925
## Est_Conc    53  0.5059245  0.3200834   0.45 0.04   1.33 0.25  0.70  0.7265159
##                Kurtosis
## Alcalinidad -0.47053491
## PH          -0.62396380
## Calcio       0.03541296
## Clorofila    1.13948206
## Min_Conc    -0.37295417
## Max_Conc    -0.66924897
```

```
## Est_Conc    -0.09204728
```

En donde podemos ver con más detalle todas las características de nuestras variables, como lo son los cuantiles, la media, la desviación estándar, la curtosis, el sesgo, etc.

Con estos datos, vamos a crear otro data frame con ahora los datos que nos interesan, que son los datos que pasaron la prueba de normalidad.

```
df3 = data.frame("PH"=x4, "Max_Conc"=x10)
df3
```

```
##     PH Max_Conc
## 1  6.1     1.43
## 2  5.1     1.90
## 3  9.1     0.06
## 4  6.9     0.84
## 5  4.6     1.50
## 6  7.3     0.48
## 7  5.4     0.72
## 8  8.1     0.38
## 9  5.8     1.40
## 10 6.4     1.47
## 11 5.4     0.86
## 12 7.2     0.73
## 13 7.2     1.01
## 14 5.8     2.03
## 15 7.6     0.11
## 16 8.2     0.18
## 17 8.7     0.43
## 18 7.8     1.50
## 19 5.8     1.33
## 20 6.7     1.44
## 21 4.4     0.93
## 22 6.7     0.94
## 23 6.1     0.61
## 24 6.9     2.04
## 25 5.5     0.62
## 26 6.9     1.12
## 27 7.3     0.52
## 28 4.5     1.38
## 29 4.8     0.84
## 30 5.8     0.69
## 31 7.8     0.59
## 32 7.4     0.65
## 33 3.6     1.90
## 34 4.4     1.02
## 35 7.9     0.30
## 36 7.1     0.29
## 37 6.8     0.37
## 38 8.4     0.06
## 39 7.0     0.63
## 40 7.5     1.41
## 41 7.0     0.26
## 42 6.8     0.26
## 43 5.9     1.05
## 44 8.3     0.48
```

```
## 45 6.7     1.40
## 46 6.2     0.95
## 47 6.2     1.10
## 48 8.9     0.40
## 49 4.3     1.24
## 50 7.0     0.90
## 51 6.9     0.69
## 52 5.2     0.40
## 53 7.9     0.51
```

Si volvemos a correr el test de normalidad en nuestro nuevo data frame, vamos a ver que vuelve a pasar ese test y que ahora tenemos puros datos con una distribución normal.

```
norm_test = mvn(df3, mvnTest="mardia")
norm_test$multivariateNormality
```

```
##              Test          Statistic           p value Result
## 1 Mardia Skewness   6.53855430534145 0.162377302354508    YES
## 2 Mardia Kurtosis -0.889321233851276 0.373830462900113    YES
## 3             MVN               <NA>              <NA>    YES
```

```
norm_test$univariateNormality
```

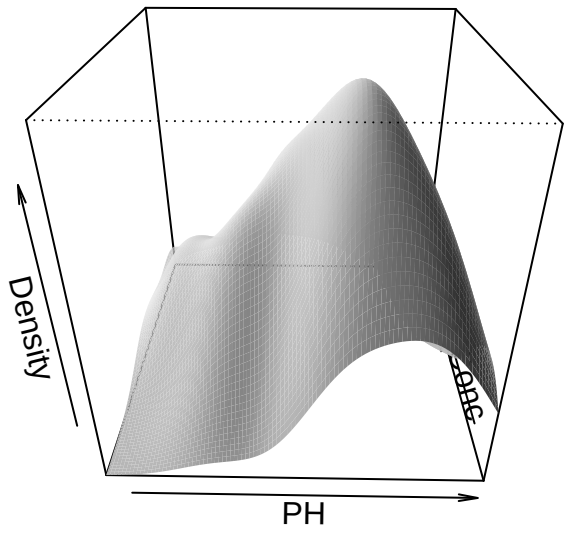
```
##               Test  Variable Statistic   p value Normality
## 1 Anderson-Darling    PH        0.3496    0.4611    YES
## 2 Anderson-Darling Max_Conc     0.6585    0.0810    YES
```

```
norm_test$Descriptives
```

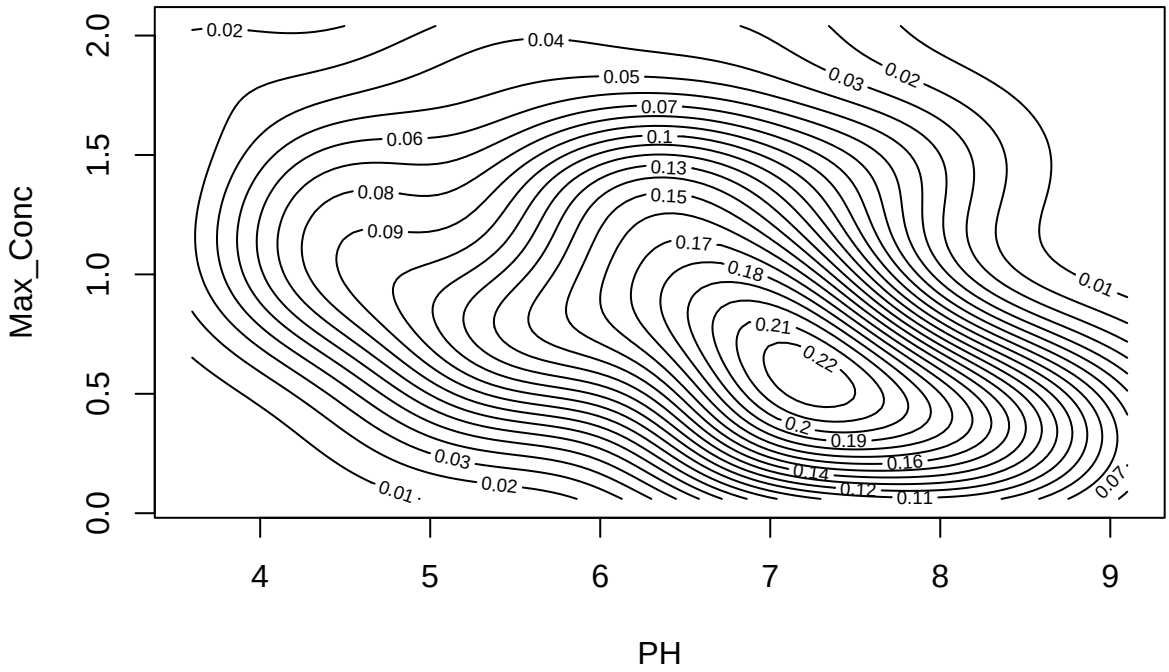
```
##           n      Mean   Std.Dev Median  Min  Max 25th 75th       Skew
## PH       53 6.5905660 1.2884493   6.80 3.60 9.10 5.80 7.40 -0.2458771
## Max_Conc 53 0.8745283 0.5220469   0.84 0.06 2.04 0.48 1.33  0.4645925
##            Kurtosis
## PH       -0.6239638
## Max_Conc -0.6692490
```

Ahora, podemos graficar los respectivos plots para ver su comportamiento bivariado.

```
perspec = mvn(df3, mvnTest="mardia", multivariatePlot="persp")
```



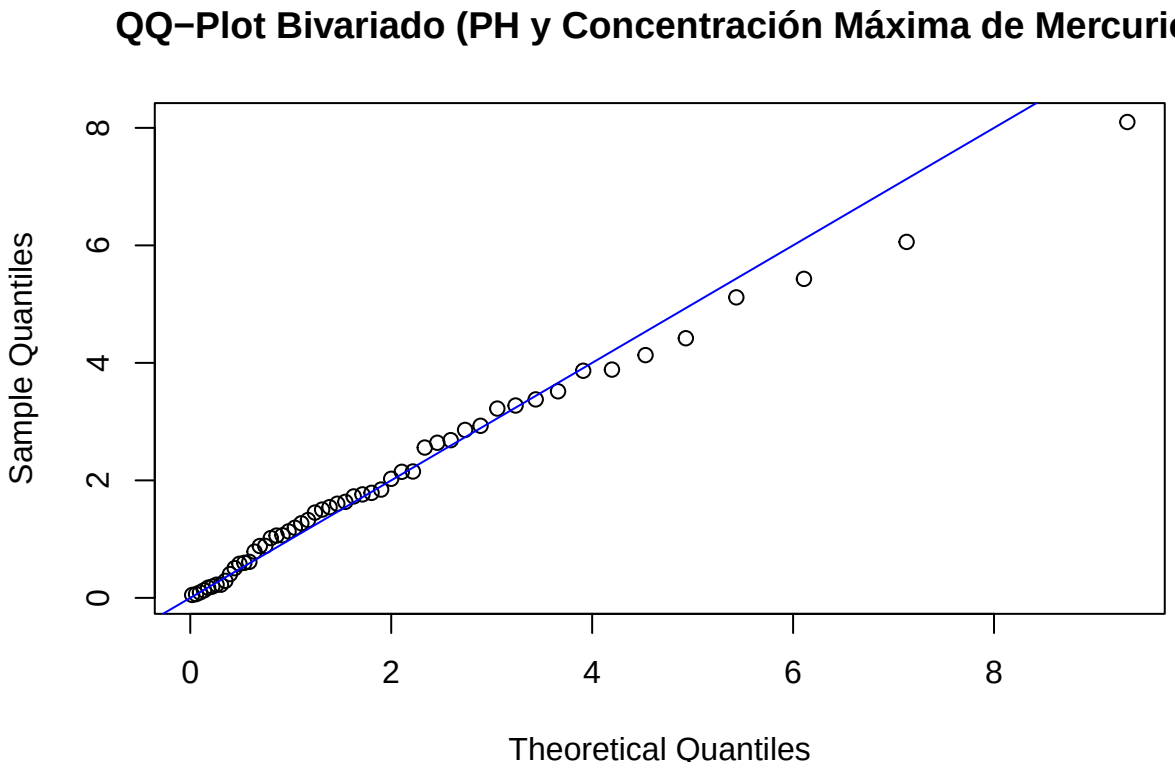
```
countour = mvn(df3, mvnTest="mardia", multivariatePlot="contour")
```



Como podemos observar en el plot del contorno, se apreciaria que los datos están centrados en que entre mayor PH tenga el agua, la concentración máxima de mercurio va a ir disminuyendo, aunque su comportamiento no es del todo homogéneo, como podemos observar.

Después de esto, vamos a buscar los datos influyentes, por lo que vamos a utilizar un grafico QQplot multivariado, que en este caso sería bivariado y lo hacemos de la siguiente manera.

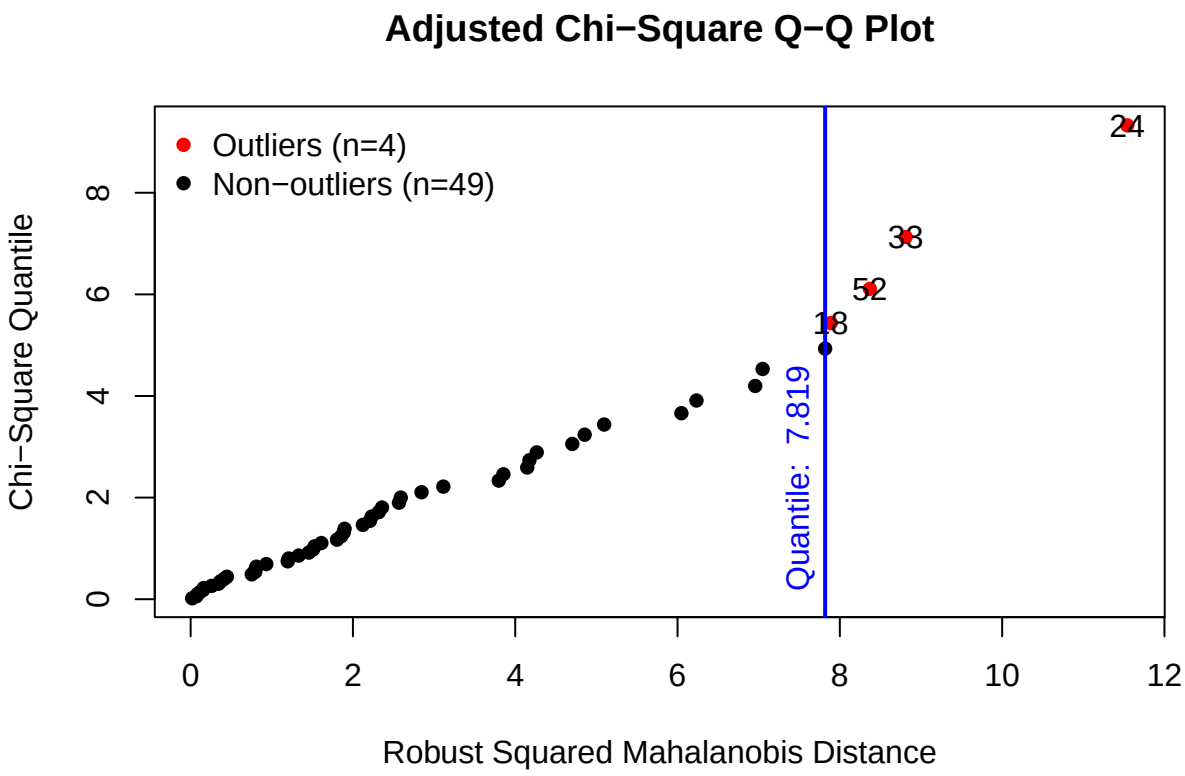
```
# Indicar que se trata de 2 variables
p = 2

# Vector de medias
X = colMeans(df3)

# Matriz de covarianza
S = cov(df3)

# Distancia de Mahalanobis
d2M =  mahalanobis(df3, X, S)

# Multinormalidad Test gráfico Q-Q Plot
plot(qchisq(((1:nrow(df3)) - 1/2)/nrow(df3), df=p), sort(d2M),
     xlab="Theoretical Quantiles", ylab="Sample Quantiles",
     main="QQ-Plot Bivariado (PH y Concentración Máxima de Mercurio)")
abline(a=0, b=1, col="blue")
```



Lo que nos dice el gráfico QQ plot bivariado, es que tiene un comportamiento con asimetría negativa es decir, que los datos están sesgados a la izquierda, por eso se comporta de la manera que sigue la tendencia normal, pero al final caen los datos.

Podemos también hacer uso de la misma librería para observar los datos atípicos, y podemos ver que solo 4 de esos datos son atípicos y que los demás se encuentran dentro de la distancia de Mahalanobis haciendo uso del **Chi-Square QQ-Plot**.

```
chi_sqr = mvn(df3, mvnTest="mardia", multivariateOutlierMethod="adj")
```



Análisis de Componentes Principales (PCA)

Para el análisis de los componentes principales, vamos a usar el data frame completo, que era **df**, pero le vamos a quitar el nombre de los lagos, ya que es una variable categórica que no influye en nada.

```
df = data.frame("Alcalinidad"=x3, "PH"=x4, "Calcio"=x5, "Clorofila"=x6,
                "Conc_Med_Merc"=x7, "Num_peces_estud"=x8, "Min_Conc"=x9,
                "Max_Conc"=x10, "Est_Conc"=x11)
df
```

```
##    Alcalinidad  PH Calcio Clorofila Conc_Med_Merc Num_peces_estud Min_Conc
## 1          5.9 6.1   3.00       0.7          1.23             6.0     0.73
## 2          3.5 5.1   1.90       3.2          1.33             7.0     0.73
## 3        116.0 9.1  44.10      79.2          0.04             6.0     0.04
## 4         39.4 6.9  16.40       3.5          0.44            12.0     0.13
## 5          2.5 4.6   2.90       1.8          1.20            12.0     0.69
## 6         19.6 7.3   4.50      44.1          0.27            14.0     0.04
## 7          5.2 5.4   2.80       3.4          0.48            10.0     0.30
## 8         71.4 8.1  55.20      33.7          0.19            12.0     0.08
## 9         26.4 5.8   9.20       1.6          0.83            37.6     0.26
## 10         4.8 6.4   4.60      22.5          0.81            12.0     0.41
## 11         6.6 5.4   2.70      14.9          0.71            12.0     0.52
## 12        16.5 7.2  13.80       4.0          0.50            12.0     0.10
## 13        25.4 7.2  25.20      11.6          0.49             7.0     0.26
## 14         7.1 5.8   5.20       5.8          1.16            37.6     0.50
## 15       128.0 7.6  77.76      79.2          0.05            11.0     0.04
## 16        83.7 8.2  66.50      79.2          0.15            10.0     0.12
## 17       108.5 8.7  35.60      79.2          0.19            37.6     0.07
## 18        61.3 7.8  57.40      13.9          0.77             6.0     0.32
```

```
## 19        6.4 5.8   4.00       4.6          1.08            10.0     0.64
## 20       31.0 6.7  15.00      17.0          0.98             6.0     0.67
## 21        7.5 4.4   2.00       9.6          0.63            12.0     0.33
## 22       17.3 6.7  10.70       9.5          0.56            12.0     0.37
## 23       12.6 6.1   3.70      21.0          0.41            12.0     0.25
## 24        7.0 6.9   6.30      32.1          0.73            12.0     0.33
## 25       10.5 5.5   6.30       1.6          0.34            10.0     0.25
## 26       30.0 6.9  13.90      21.5          0.59            37.6     0.23
## 27       55.4 7.3  15.90      24.7          0.34            10.0     0.17
## 28        3.9 4.5   3.30       7.0          0.84             8.0     0.59
## 29        5.5 4.8   1.70      14.8          0.50            11.0     0.31
## 30        6.3 5.8   3.30       0.7          0.34            10.0     0.19
## 31       67.0 7.8  58.60      43.8          0.28            10.0     0.16
## 32       28.8 7.4  10.20      32.7          0.34            10.0     0.16
## 33        5.8 3.6   1.60       3.2          0.87            12.0     0.31
## 34        4.5 4.4   1.10       3.2          0.56            13.0     0.25
## 35      119.1 7.9  38.40      16.1          0.17            12.0     0.07
## 36       25.4 7.1   8.80      45.2          0.18            13.0     0.09
## 37      106.5 6.8  77.76      16.5          0.19            13.0     0.05
## 38       53.0 8.4  45.60      79.2          0.04             6.0     0.04
## 39        8.5 7.0   2.50      12.8          0.49            12.0     0.31
## 40       87.6 7.5  77.76      20.1          1.10            10.0     0.73
## 41      114.0 7.0  72.60       6.4          0.16            14.0     0.04
## 42       97.5 6.8  45.50       6.2          0.10            12.0     0.05
## 43       11.8 5.9  24.20       1.6          0.48            10.0     0.27
## 44       66.5 8.3  26.00      79.2          0.21            12.0     0.05
## 45       16.0 6.7  41.20      24.1          0.86            12.0     0.36
## 46        5.0 6.2  23.60       9.6          0.52            12.0     0.31
## 47       25.6 6.2  12.60      27.7          0.65            37.6     0.30
## 48       81.5 8.9  20.50       9.6          0.27             6.0     0.04
## 49        1.2 4.3   2.10       6.4          0.94            10.0     0.59
## 50       34.0 7.0  13.10       4.6          0.40            12.0     0.08
## 51       15.5 6.9   5.20      16.5          0.43            11.0     0.23
## 52       17.3 5.2   3.00       2.6          0.25            12.0     0.15
## 53       71.8 7.9  20.50       8.8          0.27            12.0     0.15
##    Max_Conc Est_Conc
## 1      1.43    1.144
## 2      1.90    1.330
## 3      0.06    0.040
## 4      0.84    0.440
## 5      1.50    1.330
## 6      0.48    0.250
## 7      0.72    0.450
## 8      0.38    0.160
## 9      1.40    0.720
## 10     1.47    0.810
## 11     0.86    0.710
## 12     0.73    0.510
## 13     1.01    0.540
## 14     2.03    1.000
## 15     0.11    0.050
## 16     0.18    0.150
## 17     0.43    0.190
## 18     1.50    0.490
```

```
## 19     1.33    1.020
## 20     1.44    0.700
## 21     0.93    0.450
## 22     0.94    0.590
## 23     0.61    0.410
## 24     2.04    0.810
## 25     0.62    0.420
## 26     1.12    0.530
## 27     0.52    0.310
## 28     1.38    0.870
## 29     0.84    0.500
## 30     0.69    0.470
## 31     0.59    0.250
## 32     0.65    0.410
## 33     1.90    0.870
## 34     1.02    0.560
## 35     0.30    0.160
## 36     0.29    0.160
## 37     0.37    0.230
## 38     0.06    0.040
## 39     0.63    0.560
## 40     1.41    0.890
## 41     0.26    0.180
## 42     0.26    0.190
## 43     1.05    0.440
## 44     0.48    0.160
## 45     1.40    0.670
## 46     0.95    0.550
## 47     1.10    0.580
## 48     0.40    0.270
## 49     1.24    0.980
## 50     0.90    0.310
## 51     0.69    0.430
## 52     0.40    0.280
## 53     0.51    0.250
```

Hecho esto, ahora sí podemos hacer un análisis de componentes principales. Lo primero que tenemos que hacer es sacar la matriz de correlación de nuestras variables.

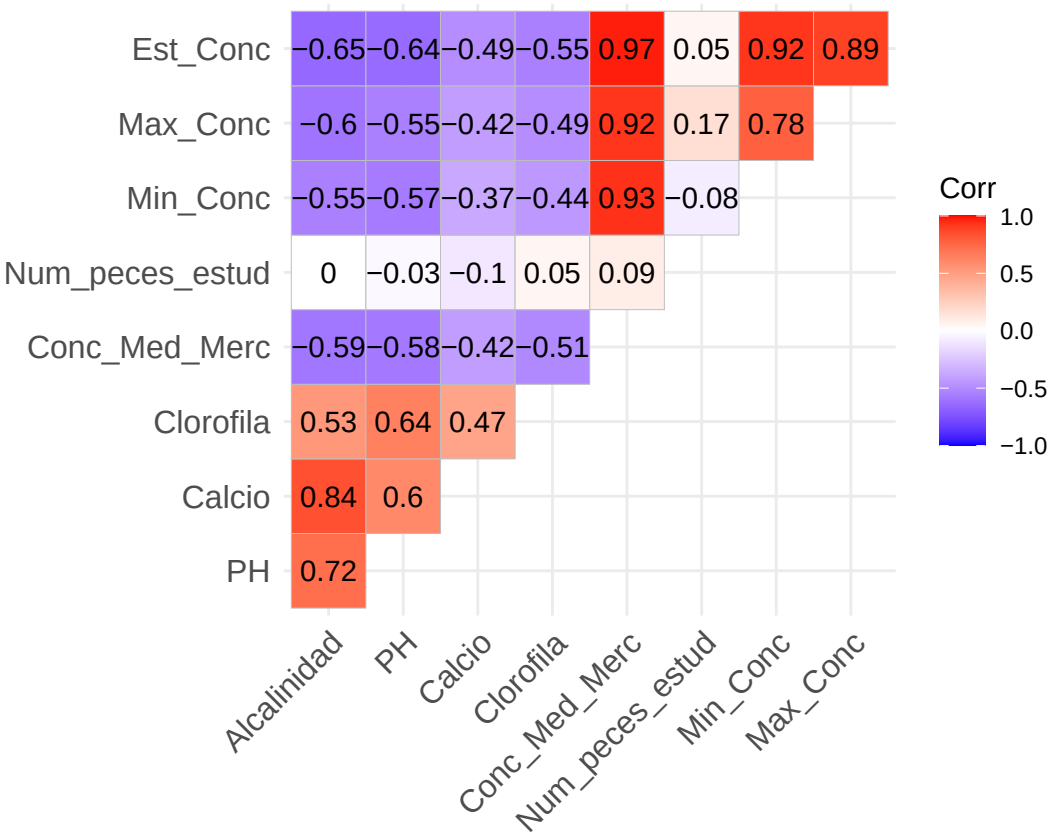
```
cor_M = cor(df)
cor_M
```

```
##                  Alcalinidad          PH      Calcio   Clorofila Conc_Med_Merc
## Alcalinidad      1.000000000  0.71916568  0.83873012  0.53291042   -0.59389671
## PH               0.719165682  1.00000000  0.60032308  0.64214187   -0.57540012
## Calcio           0.838730119  0.60032308  1.00000000  0.47059529   -0.41513174
## Clorofila        0.532910423  0.64214187  0.47059529  1.00000000   -0.51093638
## Conc_Med_Merc   -0.593896709 -0.57540012 -0.41513174 -0.51093638    1.00000000
## Num_peces_estud  0.004950691 -0.02907693 -0.09791819  0.05119524    0.08567608
## Min_Conc        -0.551113015 -0.56603763 -0.36699878 -0.44069708    0.93036718
## Max_Conc        -0.604795581 -0.55181523 -0.41843501 -0.48877230    0.91586397
## Est_Conc        -0.645338451 -0.63971986 -0.49059694 -0.54553558    0.96729866
##                 Num_peces_estud    Min_Conc   Max_Conc   Est_Conc
## Alcalinidad         0.004950691 -0.55111301 -0.6047956 -0.6453385
## PH                 -0.029076933 -0.56603763 -0.5518152 -0.6397199
```

```
## Calcio             -0.097918189 -0.36699878 -0.4184350 -0.4905969
## Clorofila           0.051195243 -0.44069708 -0.4887723 -0.5455356
## Conc_Med_Merc       0.085676083  0.93036718  0.9158640  0.9672987
## Num_peces_estud     1.000000000 -0.07893672  0.1662619  0.0528902
## Min_Conc           -0.078936722  1.00000000  0.7766115  0.9158575
## Max_Conc            0.166261906  0.77661153  1.0000000  0.8851661
## Est_Conc            0.052890202  0.91585751  0.8851661  1.0000000
```

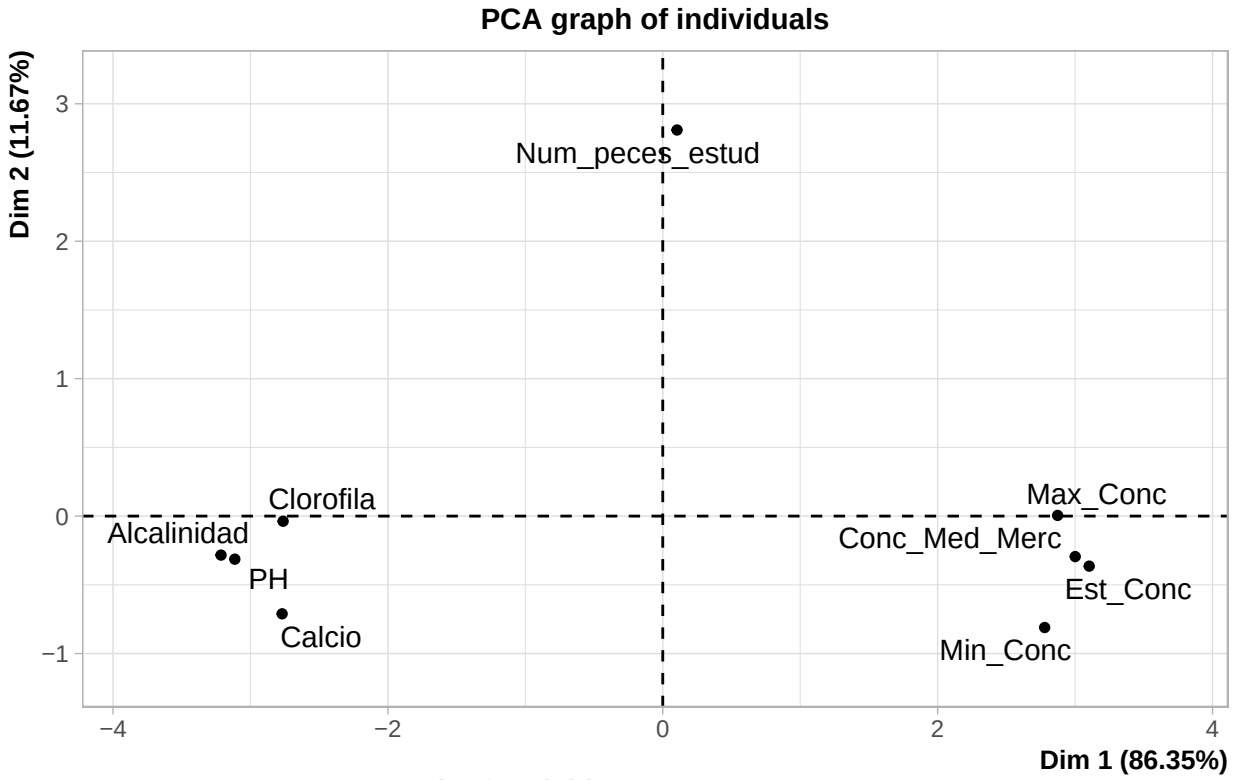
Que la verdad se ve mucho más interpretable si la graficamos.

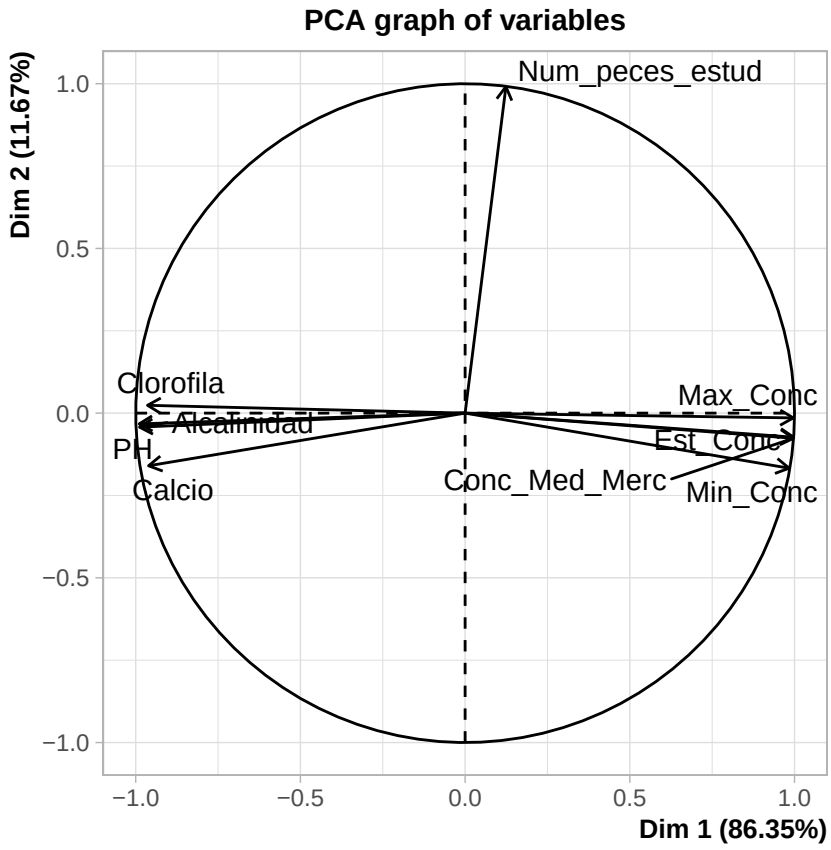
```
ggcorrplot(cor_M, lab=TRUE, type="upper")
```



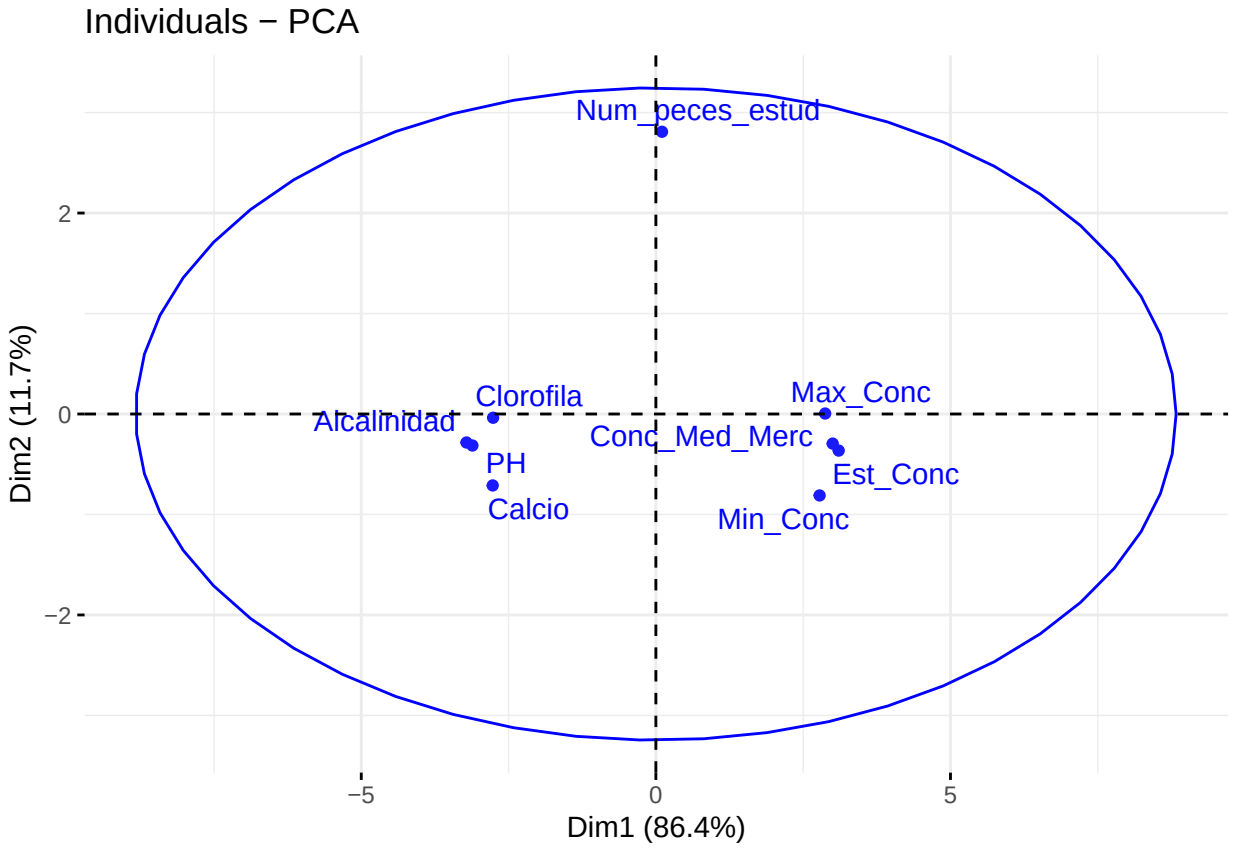
Ya desde aquí podemos hacer algunas inferencias en los datos y cuales son los componentes que más aportan y cuales no aportan, pero además de la pura matriz de correlación, vamos a hacer todo el análisis de componentes principales.

```
datos = cor_M
cp = PCA(datos)
```



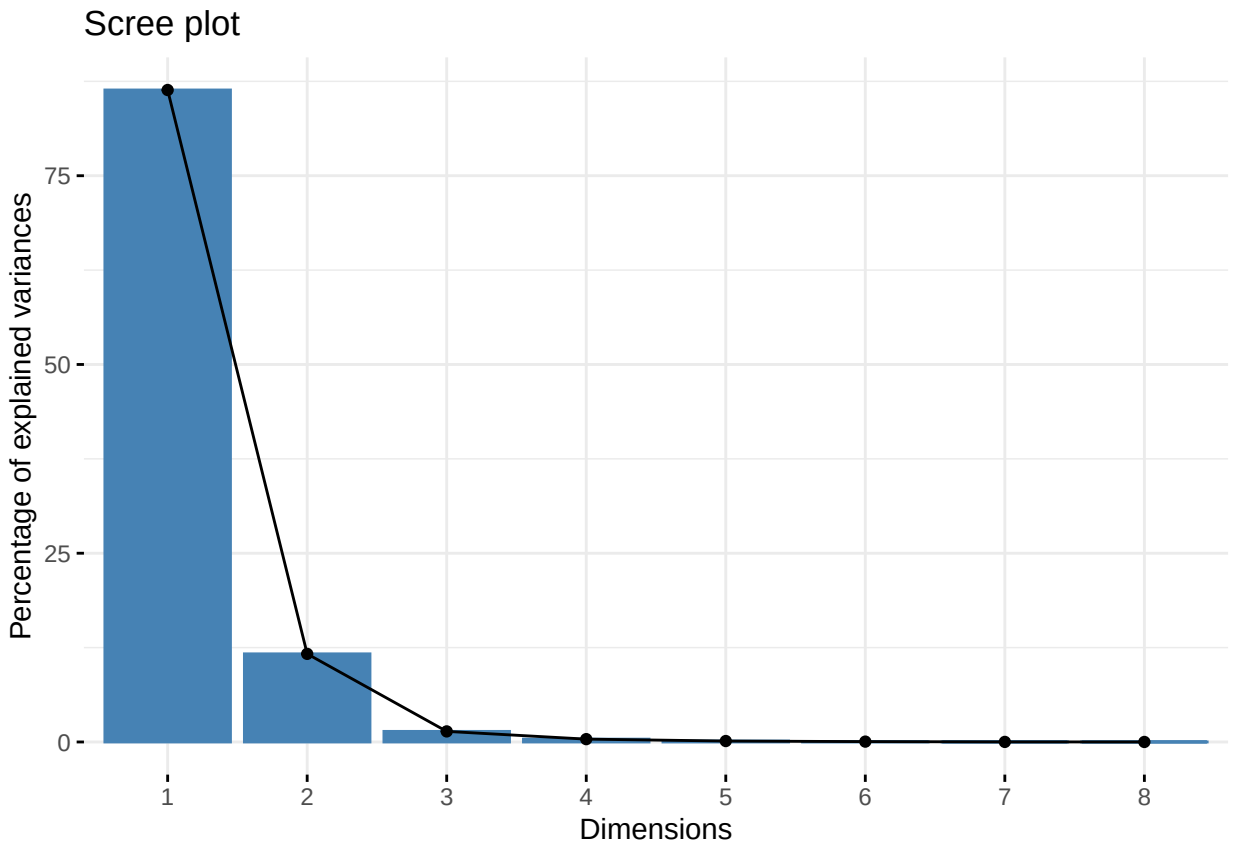



```
fviz_pca_ind(cp, col.ind="blue", addEllipses=TRUE, repel=TRUE)
```



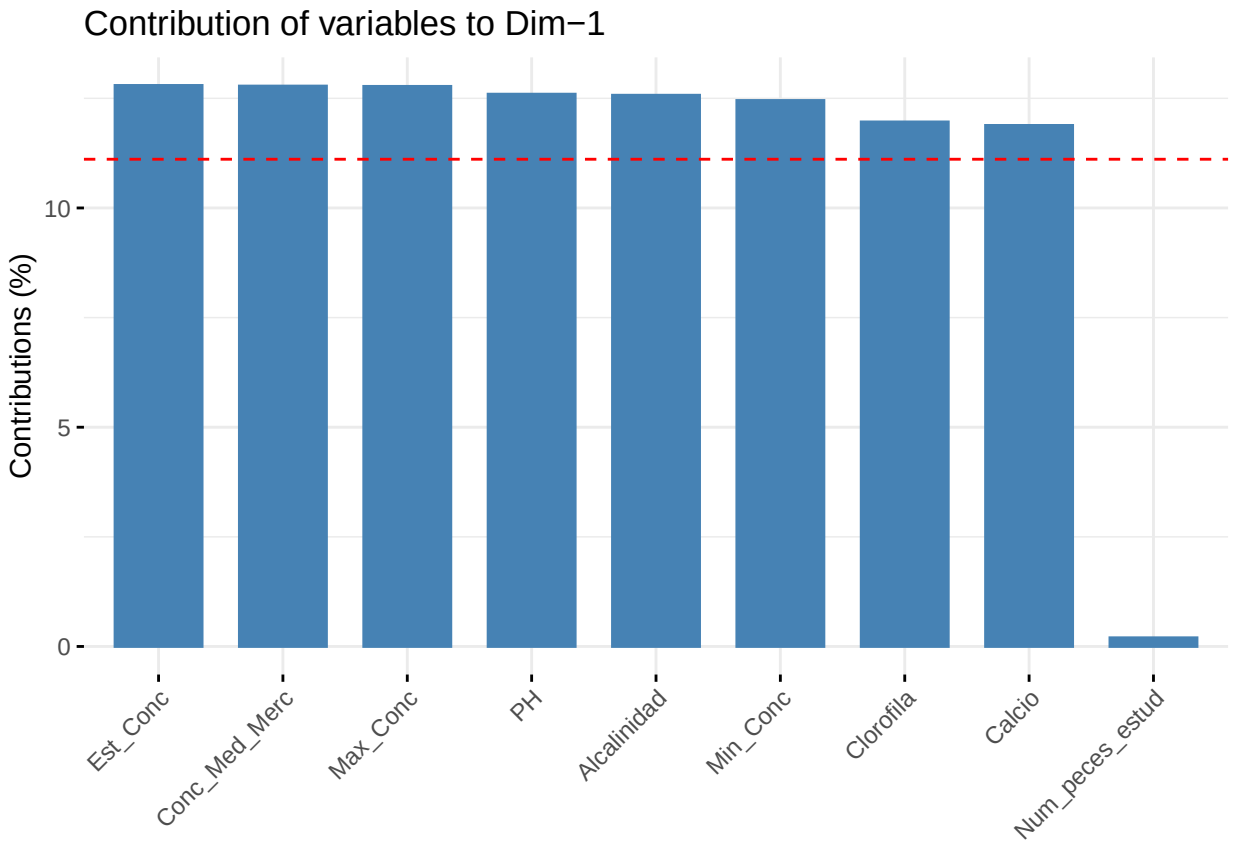
Como podemos ver en los diferentes plots pasados, lo que nos está representando es el comportamiento de las variables en base a la dimensión en la que se encuentran y su aportación a la misma. Como podemos ver, en la dimensión 1, que es nuestr PCA 1, nos representa que ahí se encuentran la mayoría de los datos, y que las que aportan positivamente son las mismas variables de mercurio, y las que representan de manera negativa, son las demás, como la clorofila, el PH, el calcio y la alcalinidad. Esto nos quiere decir y comprobar lo que hemos estado analizando en todo este estudio, que los componentes del ***PH, Calcio, Alcalinidad, y Clorofila*** nos ayudan a disminuir la cantidad de Mercurio en los peces y en el agua de los lagos.

```
fviz_screeplot(cp)
```



En el caso del plot de codo, podemos ver como nuestro problema pasa de tener 8 dimensiones, a tener solo 1 dimensión, la cual tiene una combinación lineal de las distintas variables antes presentadas.

```
fviz_contrib(cp, choice = c("var"))
```



Como mencionamos anteriormente, nuestro problema se volvió de una sola dimensión, y podemos observar que la mayoría de las variables aportan muchísimo a ese nuevo componente principal, excepto la variable de número de peces estudiados, la cual podemos ver que es obsoleta y no aporta nada de información en nuestro componente.

Si queremos, podemos observar numéricamente cuánto aporta cada componente, lo podemos hacer de la siguiente manera.

```
cp$eig
```

```
##          eigenvalue percentage of variance cumulative percentage of variance
## comp 1 7.7718853411           86.354281568                            86.35428
## comp 2 1.0504777095           11.671974550                            98.02626
## comp 3 0.1263120373            1.403467081                            99.42972
## comp 4 0.0340235915            0.378039906                            99.80776
## comp 5 0.0114337357            0.127041508                            99.93480
## comp 6 0.0045140638            0.050156265                            99.98496
## comp 7 0.0010311696            0.011457440                            99.99642
## comp 8 0.0003223515            0.003581684                           100.00000
```

En donde podemos ver que nuestro primer componente contiene el 86.354% de la información, lo cual es bastante si consideramos la reducción de la dimensionalidad. Podemos también observar que si convertimos nuestro problema a uno de 2 dimensiones, lo cual sería lo mejor, nuestra información explicativa sube a un 98%.

```
cp$var$coord
```

```
##                    Dim.1       Dim.2         Dim.3        Dim.4        Dim.5
## Alcalinidad    -0.9884036 -0.04250333  0.1382199818  0.008225411 -0.031929873
## PH             -0.9893612 -0.03272600 -0.0491671405 -0.126182982  0.025535915
```

```
## Calcio          -0.9609520 -0.15975640  0.2060666744  0.078050824  0.042473742
## Clorofila       -0.9641632  0.02377281 -0.2439011753  0.099470457  0.019212048
## Conc_Med_Merc    0.9965879 -0.07634862  0.0195621552  0.007862443  0.011472305
## Num_peces_estud  0.1239114  0.99105598  0.0458415885  0.014493631  0.001851840
## Min_Conc         0.9836662 -0.16716874  0.0007530729  0.040503408 -0.034108196
## Max_Conc         0.9963111 -0.01446895  0.0170487803 -0.005502983  0.079212399
## Est_Conc         0.9971420 -0.07305563  0.0079530751  0.010236888 -0.003999753
```

Con la ayuda de **Coord** podemos ver como nos queda nuestra nueva combinación lineal de las variables, las cuales nos dan la información de nuestros componentes principales.

```
cp$var$contrib
```

```
##                      Dim.1       Dim.2        Dim.3       Dim.4       Dim.5
## Alcalinidad     12.5702028  0.17197254 1.512505e+01  0.19885433  8.91674292
## PH              12.5945715  0.10195279 1.913838e+00 46.79736714  5.70314885
## Calcio          11.8816560  2.42957163 3.361792e+01 17.90502083 15.77803454
## Clorofila       11.9611991  0.05379898 4.709589e+01 29.08091520  3.22819069
## Conc_Med_Merc   12.7792349  0.55490106 3.029623e-01  0.18169160  1.15110041
## Num_peces_estud  0.1975586 93.49955239 1.663698e+00  0.61741085  0.02999291
## Min_Conc        12.4499927  2.66025508 4.489824e-04  4.82173110 10.17488150
## Max_Conc        12.7721368  0.01992908 2.301134e-01  0.08900537 54.87798873
## Est_Conc        12.7934475  0.50806646 5.007552e-02  0.30800358  0.13991946
```

Como podemos ver, cada variable nos da entre el 11% y 12% de la información a nuestro componente principal en la primera dimensión. Por otra parte, gracias a estos datos nos podemos dar cuenta que nuestro segundo componente, es decir nuestra dimensión 2, tiene el 93% de la información, por lo que acapara todo este componente.

Conclusión

- El test de normalidad que más nos ayudó a obtener resultados del comportamiento normal de los datos, fue el de Anderson-Darling ya que este tiende a ser más efectivo a la hora de detectar las desviaciones que se presentan en las colas de la distribución, además de que los test de normalidad se basan en la simetría y la curtosis para corroborar la misma.
- El Análisis de Componentes Principales nos ayudó para poder reducir la dimensión de nuestro problema, ya que al principio contábamos con 8 variables, que eso representa 8 diferentes dimensiones en las que las variables se pueden comportar, entonces, lo que hace el PCA, es hacer una combinación lineal de esas variables para poder reducir el tamaño de dimensiones y facilitar el procesamiento. En nuestro caso, podemos ver que los componentes del ***PH, Calcio, Alcalinidad, y Clorofila*** nos ayudan a disminuir la cantidad de Mercurio en los peces y en el agua de los lagos, mientras que el número de peces estudiados si afecta en el mismo análisis del estudio, pero en tan solo un 11.67% de la información.