Mahalanobis Distance

Sergio Adrian Ortiz Ortega

2022-05-21

Distancias de mahalanobis y gauss de DIEGO CALVO

Primero que nada cargamos los datos

```
ventas= c( 1054, 1057, 1058, 1060, 1061, 1060, 1061, 1062, 1062, 1064, 1062, 1062, 1064, 1056, 1066, 10
clientes= c(63, 66, 68, 69, 68, 71, 70, 70, 71, 72, 72, 73, 73, 75, 76, 78)
```

Utilizamos la función data.frame() para crear un juego de datos en R

```
datos <- data.frame(ventas ,clientes)
```

Generar un vector boleano indicando los valores que esten a una distancia de más de 2 desviaciones estándar de la media

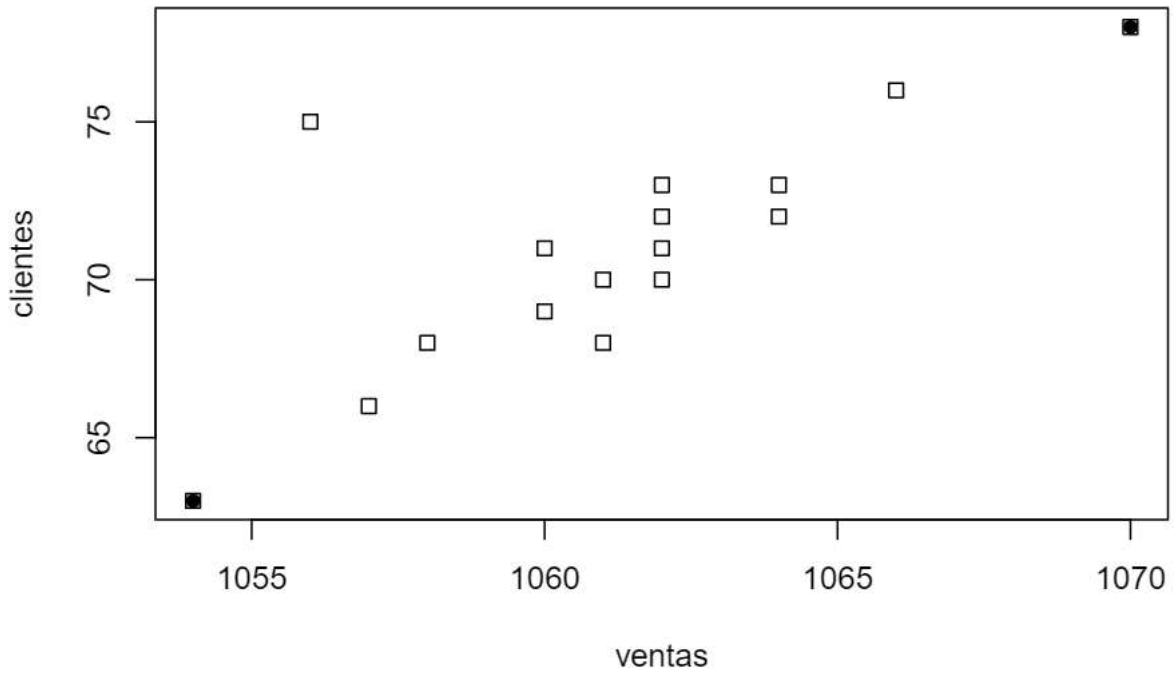
```
ventas.outlier <- abs(scale(datos$ventas)) > 2
clientes.outlier <- abs(scale(datos$clientes)) > 2
```

Almacenar los outlier encontrados para poder mostrarlos graficamente

```
outlier <- rbind(datos[ventas.outlier ,], datos[clientes.outlier ,])
```

Visualizar el gráfico con los datos destacando sus outlier

```
plot(datos, pch=0)
points(outlier , pch=16)
```



Método de distancia Mahalanobis Determinar el número de outlier que queremos encontrar

```
num.outliers <- 2
```

Ordenar los datos de mayor a menor distancia, según la métrica de Mahalanobis.

```
mah.ordenacion <- order(mahalanobis(datos , colMeans( datos), cov(datos)), decreasing=TRUE)
```

Generar un vector boleano los dos valores más alejados segun la distancia Mahalanobis.

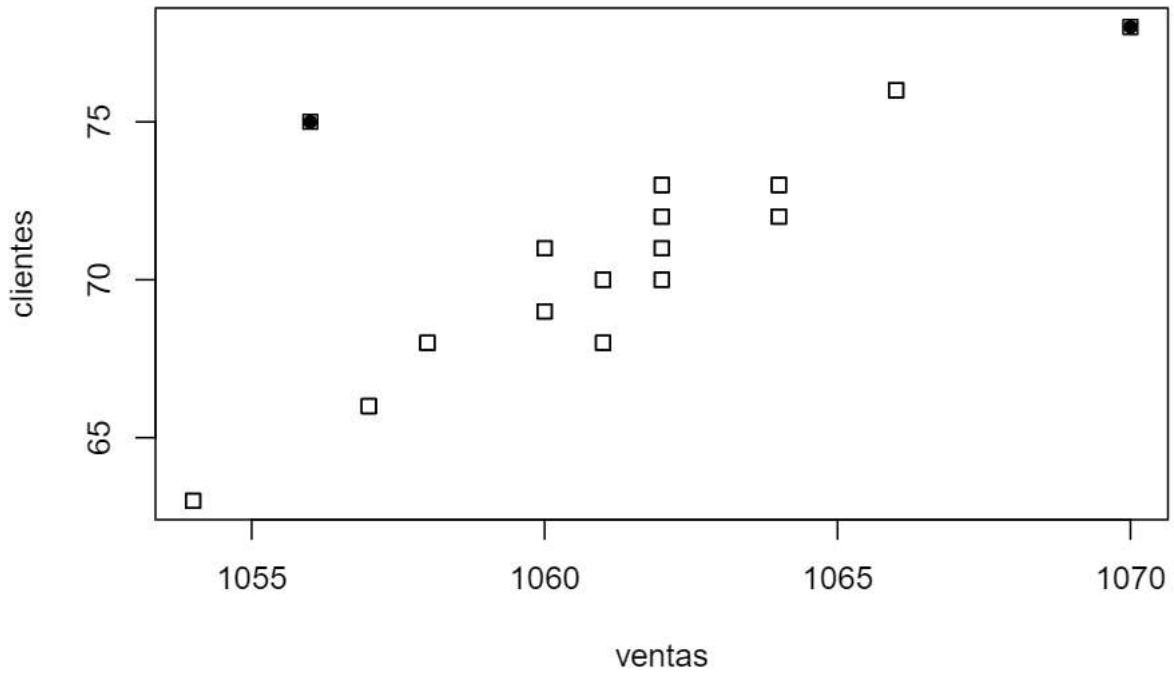
```
outlier2 <- rep(FALSE , nrow(datos))
outlier2[mah.ordenacion[1:num.outliers]] <- TRUE
```

Resaltar con un punto relleno los 2 valores outliers.

```
colorear.outlier <- outlier2 * 16
```

Visualizar el gráfico con los datos destacando sus outlier.

```
plot(datos , pch=0)
points(datos , pch=colorear.outlier)
```



Distancia de Mahalanobis ejemplo de R.

primero encendemos las librerias ya que requiere graficos

```
require(graphics)
```

hacemos la base de datos

```
ma <- cbind(1:6, 1:3)
(S <-  var(ma))
```

```
##      [,1] [,2]
## [1,]  3.5  0.8
## [2,]  0.8  0.8
```

```
mahalanobis(c(0, 0), 1:2, S)
```

```
## [1] 5.37037
```

hacemos la matrix para las variables

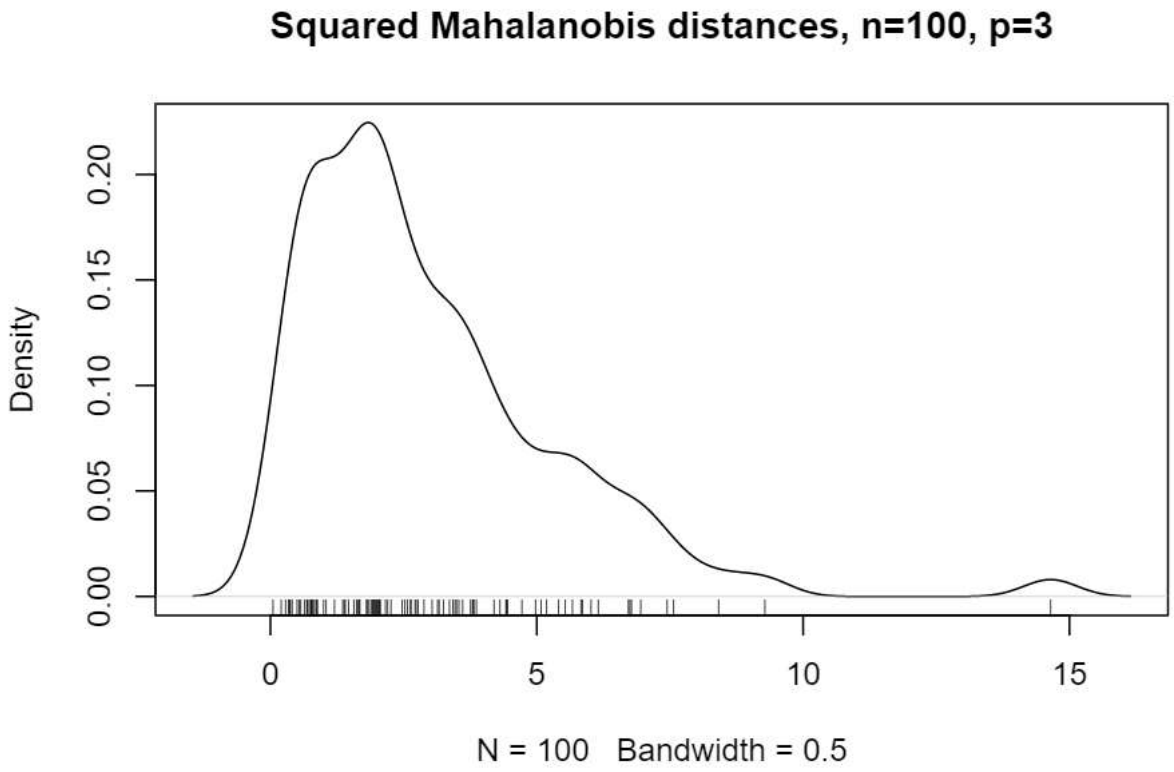
```
x <- matrix(rnorm(100*3), ncol = 3)
stopifnot(mahalanobis(x, 0, diag(ncol(x))) == rowSums(x*x))

Sx <- cov(x)
D2 <- mahalanobis(x, colMeans(x), Sx)
```

Here, D^2 = usual squared Euclidean distances

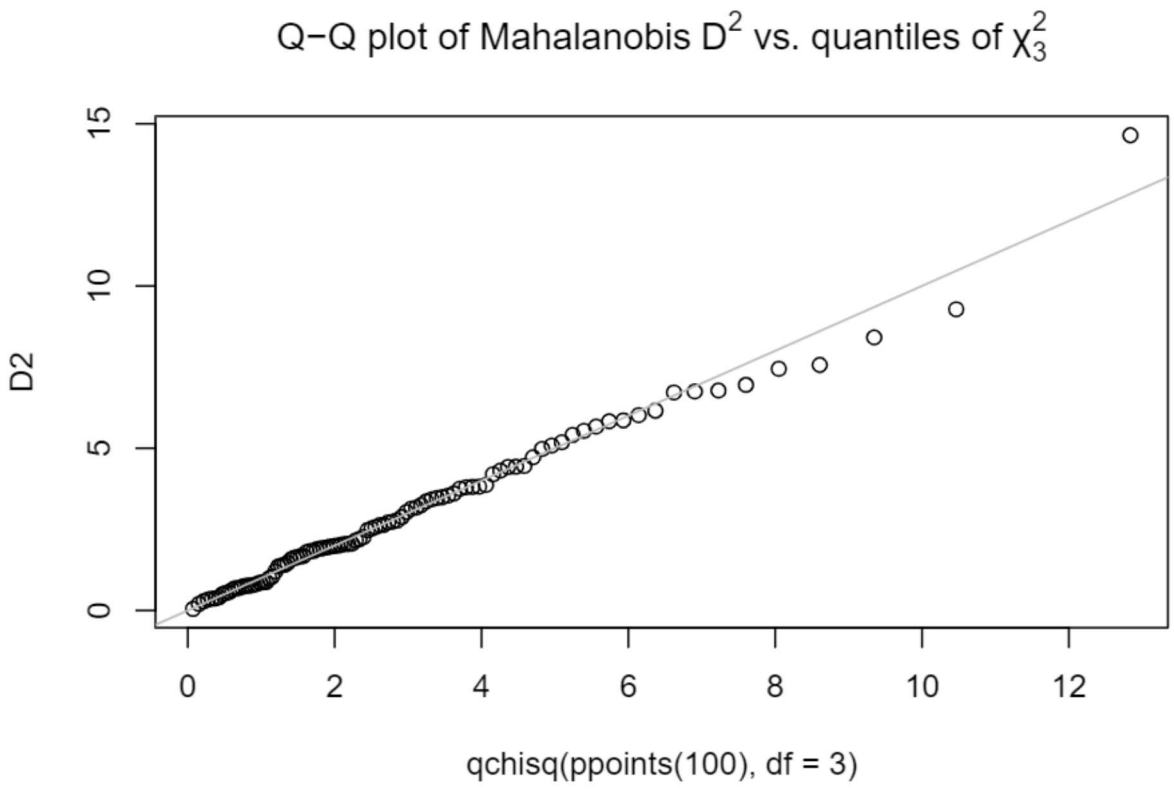
hacemos el primer grafico para ver como se conportan

```
plot(density(D2, bw = 0.5),
     main="Squared Mahalanobis distances, n=100, p=3") ; rug(D2)
```



hacemos el segundo graficos para ver su comportamiento mejor para ver si hay linealidad

```
qqplot(qchisq(ppoints(100), df = 3), D2,
       main = expression("Q-Q plot of Mahalanobis" * ~D^2 *
                         " vs. quantiles of" * ~ chi[3]^2))
abline(0, 1, col = 'gray')
```



Mahalanobis ejemplo con base propio

plantamiento del problema

es una base de datos que tiene personas con hipertension y mediremos las variables suero_creatinina,suero_sodio para notar las distancias de las variables

cargamos los datos

```
library(readxl)
tension<- read_excel("tenso.xlsx")
```

Exploracion de matriz

```
dim(tension)
```

```
## [1] 50 13
```

```
str(tension)
```

```
## tibble [50 x 13] (S3: tbl_df/tbl/data.frame)
##  $ edad                   : num [1:50] 75 55 65 50 65 90 75 60 65 80 ...
##  $ anemia                 : num [1:50] 0 0 0 1 1 1 1 1 0 1 ...
##  $ diabetes               : num [1:50] 0 0 0 0 1 0 0 1 0 0 ...
##  $ Alta_presión_sanguínea : num [1:50] 1 0 0 0 0 1 0 0 0 1 ...
##  $ sexo                   : num [1:50] 1 1 1 1 0 1 1 1 0 1 ...
##  $ fuma                   : num [1:50] 0 0 1 0 0 1 0 1 0 1 ...
##  $ MUERTE_EVENTO          : num [1:50] 1 1 1 1 1 1 1 1 1 1 ...
##  $ creatinina_fosfoquinasa: num [1:50] 582 7861 146 111 160 ...
```

```
##  $ fracción_de_eyección   : num [1:50] 20 38 20 20 20 40 15 60 65 35 ...
##  $ plaquetas              : num [1:50] 265000 263358 162000 210000 327000 ...
##  $ suero_creatinina       : num [1:50] 1.9 1.1 1.3 1.9 2.7 2.1 1.2 1.1 1.5 9.4 ...
##  $ suero_sodio            : num [1:50] 130 136 129 137 116 132 137 131 138 133 ...
##  $ tiempo                 : num [1:50] 4 6 7 7 8 8 10 10 10 10 ...
```

```
colnames(tension)
```

```
##  [1] "edad"                   "anemia"
##  [3] "diabetes"               "Alta_presión_sanguínea"
##  [5] "sexo"                   "fuma"
##  [7] "MUERTE_EVENTO"          "creatinina_fosfoquinasa"
##  [9] "fracción_de_eyección"   "plaquetas"
## [11] "suero_creatinina"       "suero_sodio"
## [13] "tiempo"
```

con esto podemos ver que no hay datos NA ## transformar las variables para que funcionen con el codigo

```
fuma<-factor(tension$fuma,
             levels= c("1","0"))
anemia<-factor(tension$anemia,
               levels= c("1","0"))
sexo<-factor(tension$sexo,
             levels= c("1","0"))
diabetes<-factor(tension$diabetes,
                 levels= c("1","0"))
Alta_presión_sanguínea<-factor(tension$Alta_presión_sanguínea,
                               levels= c("1","0"))
MUERTE_EVENTO<-factor(tension$MUERTE_EVENTO,
                      levels= c("1","0"))
edad<-as.numeric(tension$edad,strict = TRUE)
creatinina_fosfoquinasa<-as.numeric(tension$creatinina_fosfoquinasa,strict = TRUE)
fracción_de_eyección<-as.numeric(tension$fracción_de_eyección,strict = TRUE)
plaquetas<-as.numeric(tension$plaquetas,strict = FALSE)
suero_creatinina<-as.numeric(tension$suero_creatinina,strict = TRUE)
suero_sodio<-as.numeric(tension$suero_sodio,strict = TRUE)
tiempo<-as.numeric(tension$tiempo,strict = TRUE)
datos<-data.frame(suero_creatinina,suero_sodio)
```

Generar un vector boleano indicando los valores que esten a una distancia de más de 2 desviaciones estándar de la media

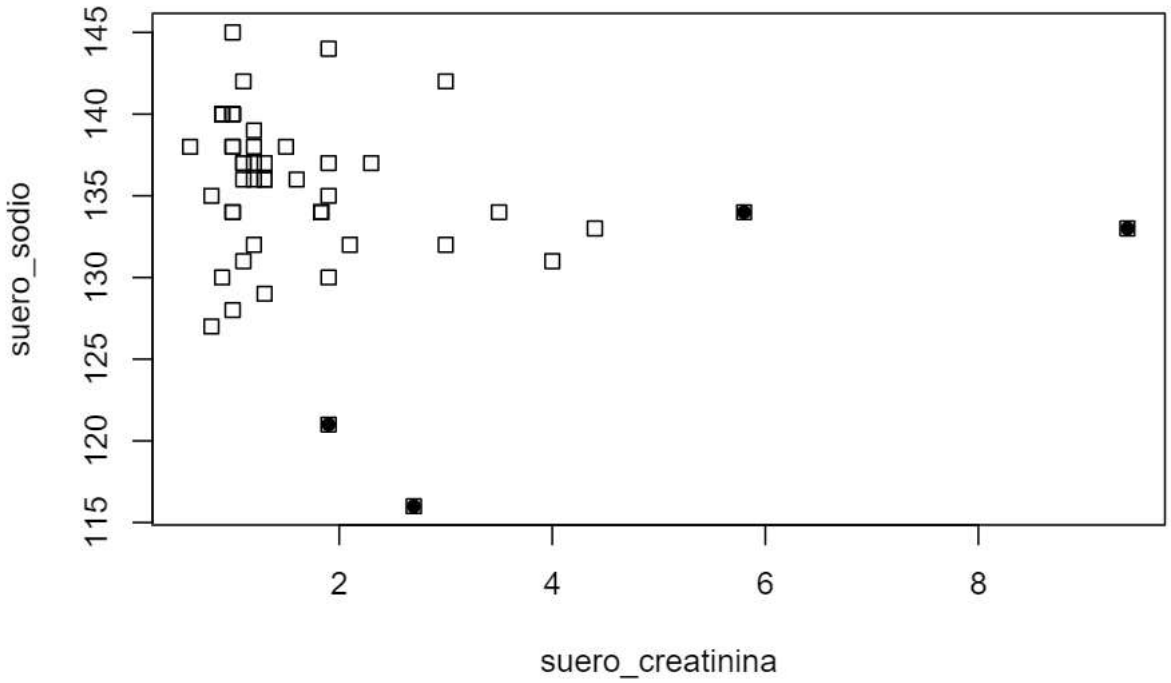
```
suero_sodio.outlier <- abs(scale(datos$suero_sodio)) > 2
suero_creatinina.outlier <- abs(scale(datos$suero_creatinina)) > 2
```

Almacenar los outlier encontrados para poder mostrarlos graficamente

```
outlier <- rbind(datos[suero_sodio.outlier ,],
                 datos[suero_creatinina.outlier ,])
```

Visualizar el gráfico con los datos destacando sus outlier

```
plot(datos, pch=0)
points(outlier , pch=16)
```



Determinar el número de outlier que queremos encontrar.

```
num.outliers <- 2
```

Ordenar los datos de mayor a menor distancia, según la métrica de Mahalanobis.

```
mah.ordenacion <- order(mahalanobis(datos , colMeans( datos),
                                    cov(datos)), decreasing=TRUE)
```

Generar un vector boleano los dos valores más alejados segun la distancia Mahalanobis.

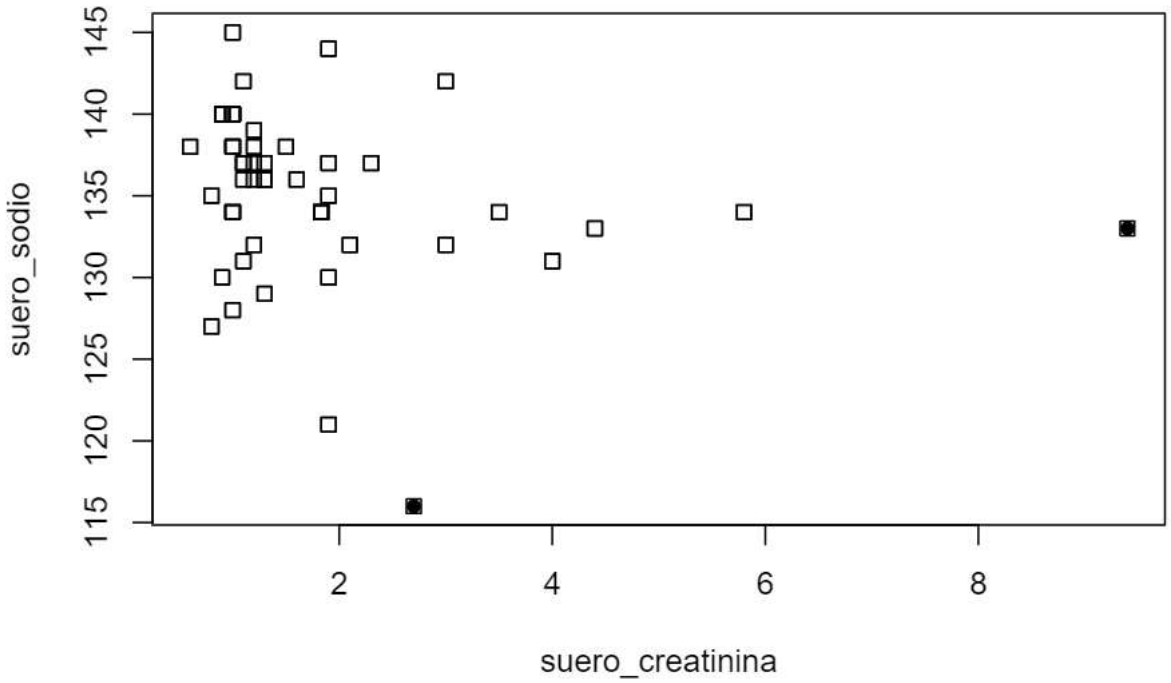
```
outlier2 <- rep(FALSE , nrow(datos))
outlier2[mah.ordenacion[1:num.outliers]] <- TRUE
```

Resaltar con un punto relleno los 2 valores outliers.

```
colorear.outlier <- outlier2 * 16
```

Visualizar el gráfico con los datos destacando sus outlier.

```
plot(datos , pch=0)
points(datos , pch=colorear.outlier)
```


suero_sodio
115
120
125
130
135
140
145
2
4
6
8
suero_creatinina