

Cálculo de la distancia de Mahalanobis

Jimena Isaura Medina Padilla

25/5/2022

Ejercicio del Calvo

1.-Cargar los datos

```
ventas= c( 1054, 1057, 1058, 1060, 1061, 1060, 1061, 1062, 1062, 1064, 1062, 1062, 1064, 1056, 1066, 1070)  
clientes= c(63, 66, 68, 69, 68, 71, 70, 70, 71, 72, 72, 73, 73, 75, 76, 78)
```

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

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

```
dim(datos)
```

```
## [1] 16  2
```

```
str(datos)
```

```
## 'data.frame':  16 obs. of  2 variables:  
## $ ventas  : num  1054 1057 1058 1060 1061 ...  
## $ clientes: num   63 66 68 69 68 71 70 70 71 72 ...
```

```
summary(datos)
```

```
##      ventas      clientes  
## Min.   :1054   Min.     :63.00  
## 1st Qu.:1060   1st Qu.:68.75  
## Median :1062   Median  :71.00  
## Mean   :1061   Mean    :70.94  
## 3rd Qu.:1062   3rd Qu.:73.00  
## Max.   :1070   Max.     :78.00
```

3.-Determinar el numero de outlier que queremos encontrar

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

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

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

```
## [1] 14 16 1 15 2 5 3 10 13 8 12 4 6 7 9 11
```

5.-Generar un vector booleano los dos valores más alejados según la distancia Mahalanobis.

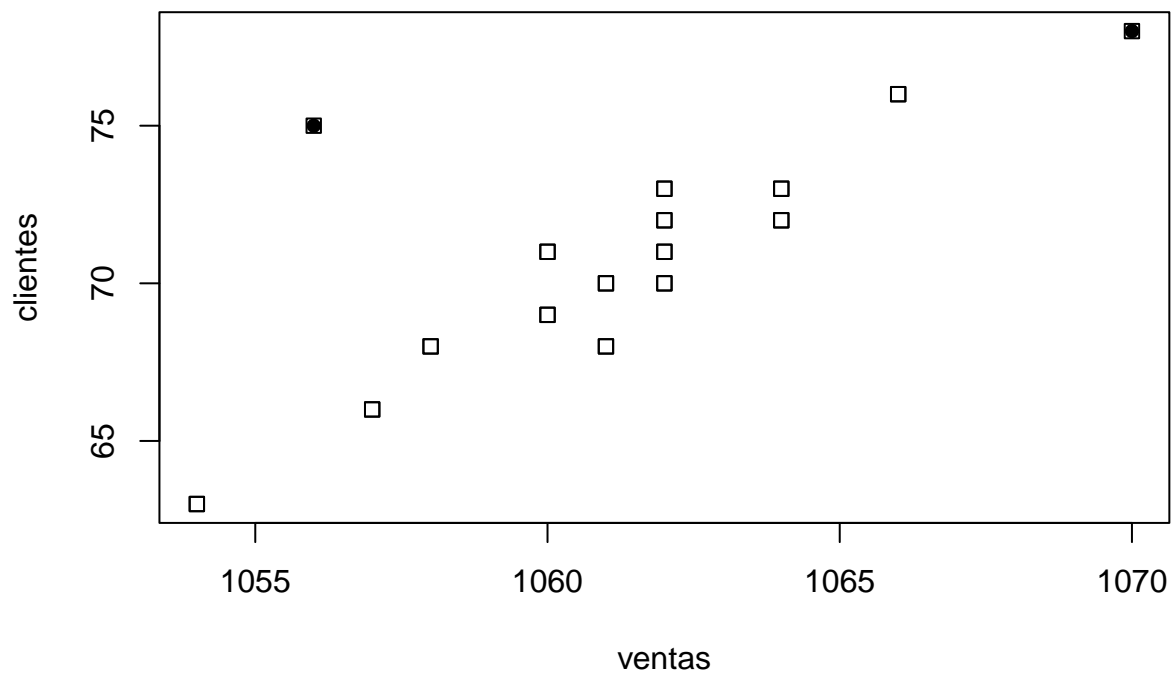
```
outlier2 <- rep(FALSE , nrow(datos))
outlier2[mah.ordenacion[1:num.outliers]] <- TRUE
```

6.-Resaltar con un punto relleno los 2 valores outliers.

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

7.-Visualizar el gráfico con los datos destacando sus outlier.

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



2º Ejercicio- Función de Mahalanobis en R (Help)

```
require(graphics)
```

```
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)
```

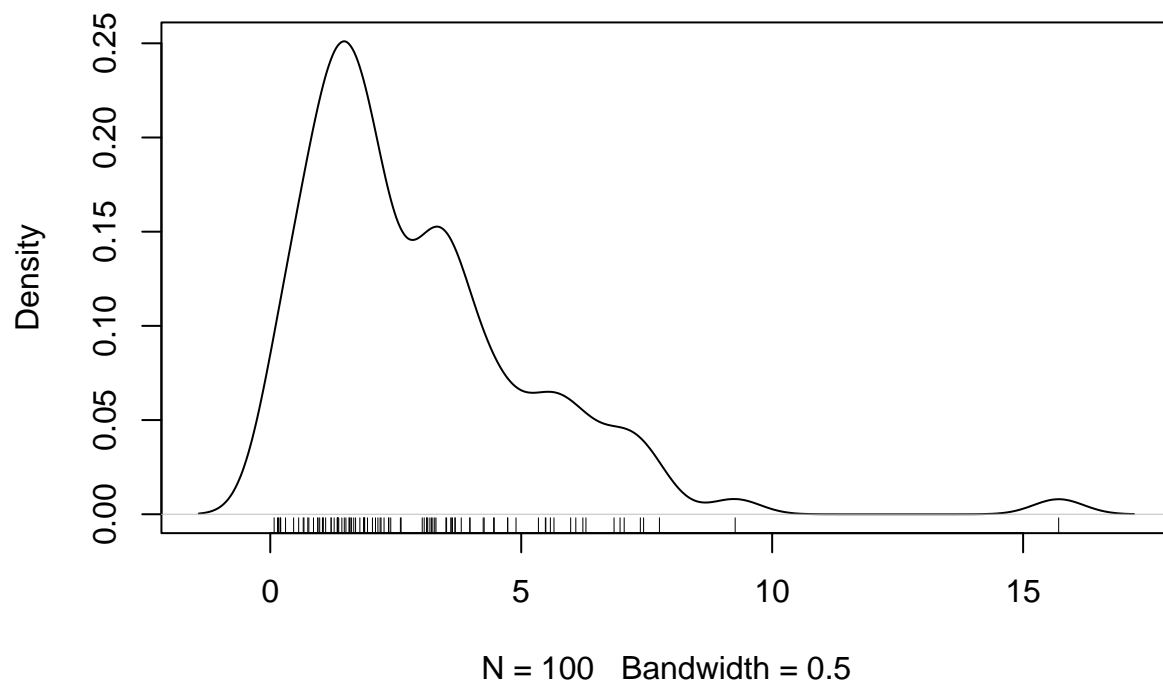
```
## [1] 5.37037
```

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

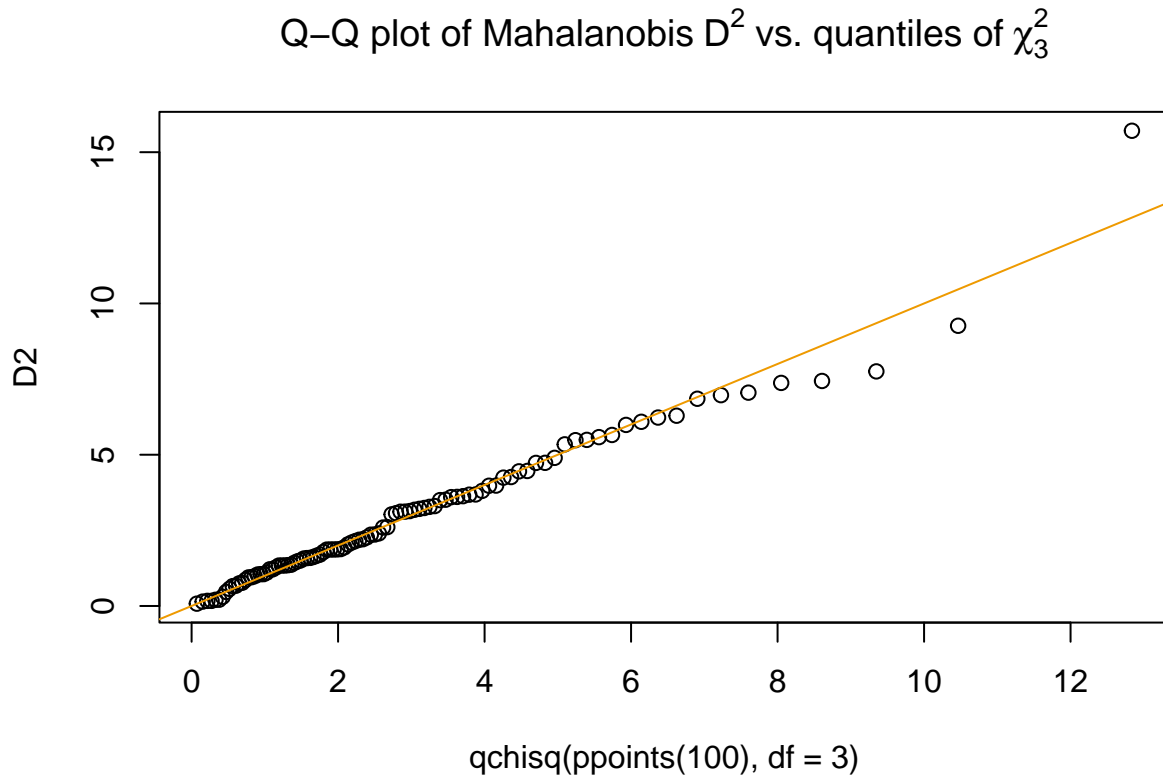
##- Here, D^2 = usual squared Euclidean distances

```
Sx <- cov(x)  
D2 <- mahalanobis(x, colMeans(x), Sx)  
plot(density(D2, bw = 0.5),  
     main="Squared Mahalanobis distances, n=100, p=3") ; rug(D2)
```

Squared Mahalanobis distances, n=100, p=3



```
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 = 'orange2')
```



Ejercicio 3 con diseño propio

1.-Cargar la libreria

```
library(datos)
```

```
## Warning: package 'datos' was built under R version 4.1.2
```

```
## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'tibble'
```

```
## Warning: replacing previous import 'lifecycle::last_warnings' by
## 'rlang::last_warnings' when loading 'pillar'
```

2.-Cargamos la base de datos

```
M<-data.frame(datos::fiel)
```

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

```
M<- data.frame(M)
```

```
dim(M)
```

```
## [1] 272 2
```

```
str(M)
```

```
## 'data.frame': 272 obs. of 2 variables:
## $ erupciones: num 3.6 1.8 3.33 2.28 4.53 ...
## $ espera : num 79 54 74 62 85 55 88 85 51 85 ...
```

```
summary(M)
```

```
## erupciones espera
## Min. :1.600 Min. :43.0
## 1st Qu.:2.163 1st Qu.:58.0
## Median :4.000 Median :76.0
## Mean :3.488 Mean :70.9
## 3rd Qu.:4.454 3rd Qu.:82.0
## Max. :5.100 Max. :96.0
```

4.-Determinar el número de outlier que queremos encontrar

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

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

```
ah.ordenacion <- order(mahalanobis(M , colMeans( M), cov(M)), decreasing=TRUE)
```

6.-Generar un vector booleano los dos valores más alejados según la distancia Mahalanobis.

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

7.-Resaltar con un punto relleno los 2 valores outliers.

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

8.-Visualizar el gráfico con los datos destacando sus outlier.

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

