Mahalanobis

Oscar Elí Bonilla Morales

2022-05-26

Distancia de Mahalanobis

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)
```

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
```

Calculo de la distancia

El método de distancia Mahalanobis mejora el método clásico de distancia de Gauss eliminando el efecto que pueden producir la correlación entre las variables a analizar

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)
mah.ordenacion
```

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

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

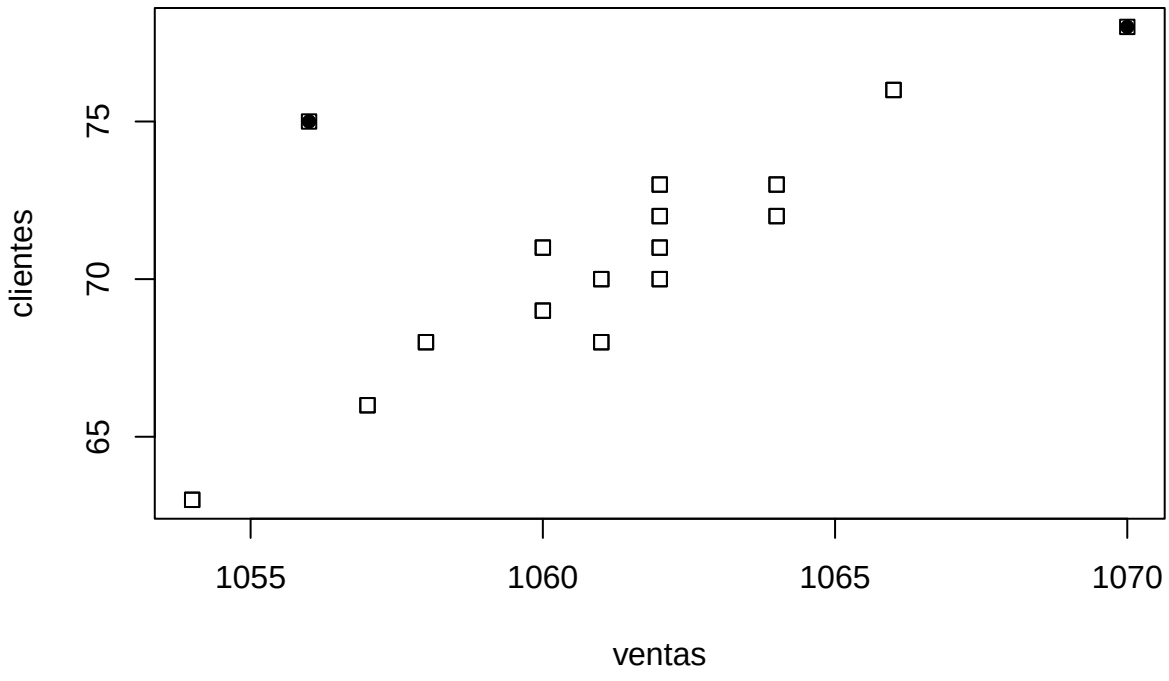
```
outlier2 <- rep(FALSE , nrow(datos))
outlier2[mah.ordenacion[1:num.outliers]] <- TRUE
```

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)
```



Ejercicio 2

```
require(graphics)

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

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

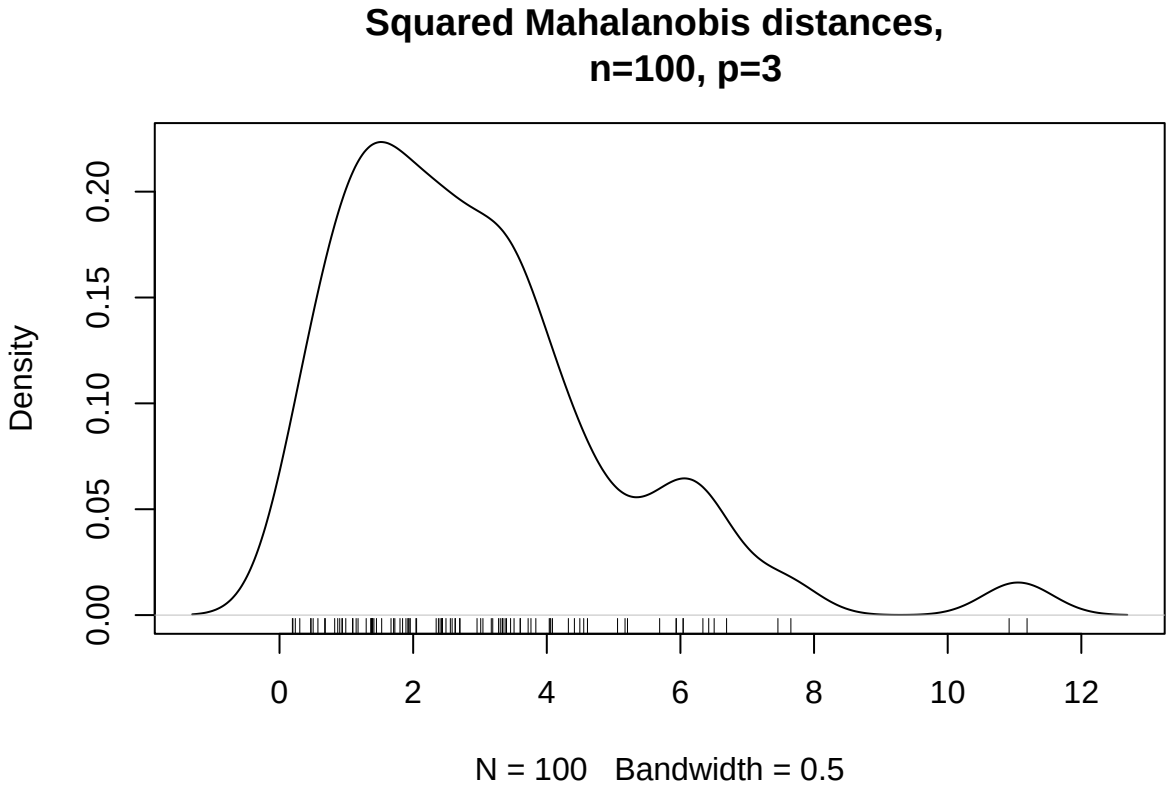
```
mahalanobis(c(0, 0), 1:2, S)
```

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

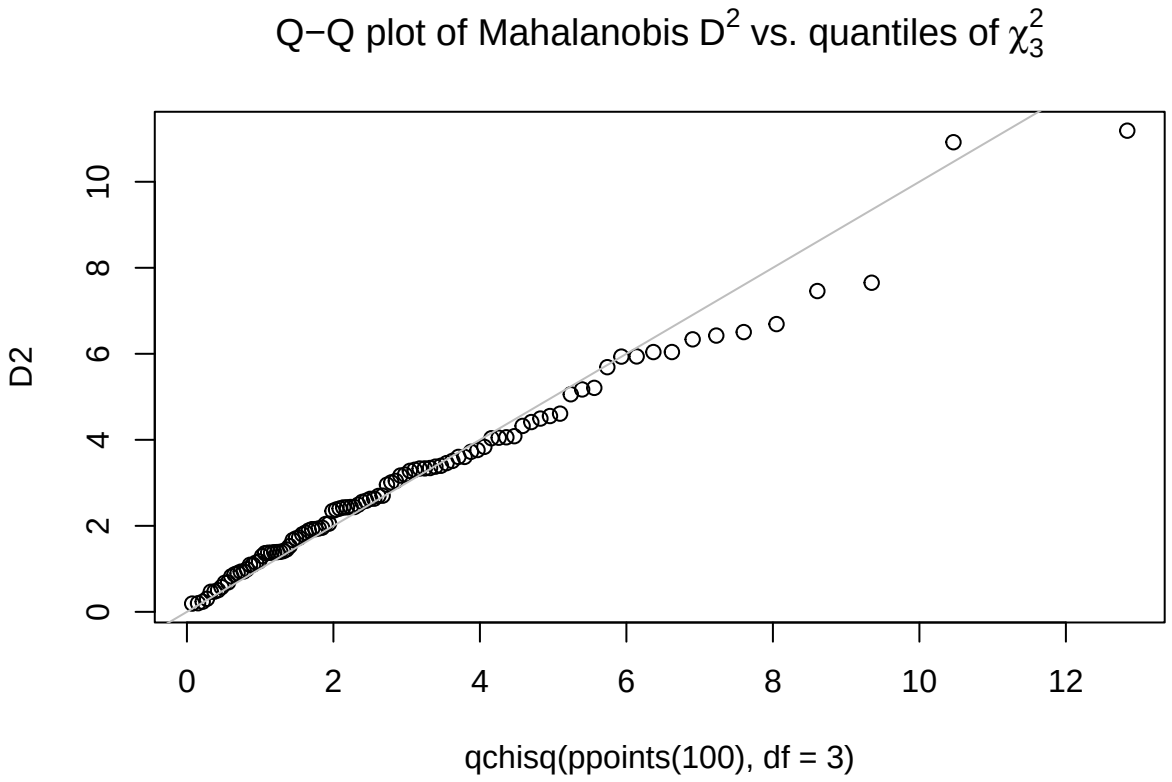
```
x <- matrix(rnorm(100*3), ncol = 3)
stopifnot(mahalanobis(x, 0,
                      diag(ncol(x))) == rowSums(x*x))
```

##- 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)
```




```
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')
```



Ejercicio 3

Diseñar un ejercicio utilizando la distancia de Mahalanobis.

Incluye:

- 1. Planteamiento del problema.
- 2. Simular los datos o utilizar una matriz Precargada en R.
- 3. Dar tu interpretacion.

Se manda a llamar la funcion "data()" para observar todos los distintos datasets

#precargados en R

```
data(trees)
head(trees)
```

```
##   Girth Height Volume
## 1   8.3     70   10.3
## 2   8.6     65   10.3
## 3   8.8     63   10.2
## 4  10.5     72   16.4
## 5  10.7     81   18.8
## 6  10.8     83   19.7
```

#Distancia de Mahalanobis

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

```
datos <- data.frame(trees)
```

```
dim(datos)
```

```
## [1] 31  3
```

```
str(datos)
```

```
## 'data.frame':    31 obs. of  3 variables:
##  $ Girth : num  8.3 8.6 8.8 10.5 10.7 10.8 11 11 11.1 11.2 ...
##  $ Height: num  70 65 63 72 81 83 66 75 80 75 ...
##  $ Volume: num  10.3 10.3 10.2 16.4 18.8 19.7 15.6 18.2 22.6 19.9 ...
```

```
summary(datos)
```

```
##      Girth           Height       Volume
##  Min.   : 8.30   Min.   :63   Min.   :10.20
##  1st Qu.:11.05   1st Qu.:72   1st Qu.:19.40
##  Median :12.90   Median :76   Median :24.20
##  Mean   :13.25   Mean   :76   Mean   :30.17
##  3rd Qu.:15.25   3rd Qu.:80   3rd Qu.:37.30
##  Max.   :20.60   Max.   :87   Max.   :77.00
```

Calculo de la distancia

El método de distancia Mahalanobis mejora el método clásico de distancia de Gauss eliminando el efecto que pueden producir la correlación entre las variables a analizar

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)
mah.ordenacion
```

```
##  [1] 31  3 20 18  2  1 28  6 17 24 26 30 27 19 29  5  7 16  9 15 22 11 25 14  4
## [26]  8 23 12 10 13 21
```

Una vez obtenidos los resultados podemos decir que nuestros datos numero 31, 2, 20 son aquellos que tienen mayor distancia de mahalanobis, mientras que en los datos numero 10, 12, y 21 es menor

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)
```