Distancia de Mahalanobis

Lino Oswaldo Sánchez Juarez

5/6/2022

Introducción

Su utilidad radica en que es una forma de determinar la similitud entre dos variables aleatorias multidimensionales. Se diferencia de la distancia euclídea en que tiene en cuenta la correlación entre las variables aleatorias.

Uno de los puntos fuertes es que es invariante ante los cambios de escala y no depende de las unidades de medida, esto quiere decir que nuestras variables no deben medir lo mismo ni estar en la misma unidad de medida, lo que la convierte en una distancia muy pragmática para aplicar en muchos.

Cargar los datos

Para este ejercicio usaremos datos capturados en vectrores de un ejercicio extraido del repertorio de Diego Calvo, sobre las ventas de una empresa.

```
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)
```

Los convertimos a data freme

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

Cálculo 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
```

Como es un estudio con autlier determinamos cuantos serán y a partir de aquí se calculara la distancia

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
```

Ordenamos las distancias de mahalanobis de los datos, las medias de las columnas y la covarianza de los datos y ordenados de mayor a menor; lo visualizamos para observar los datos, donde observamos que los datos 14,16 y 1 las distancias de mahalanobis son mayor y en los datos 7, 9 y 11 las distancias son menores.

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

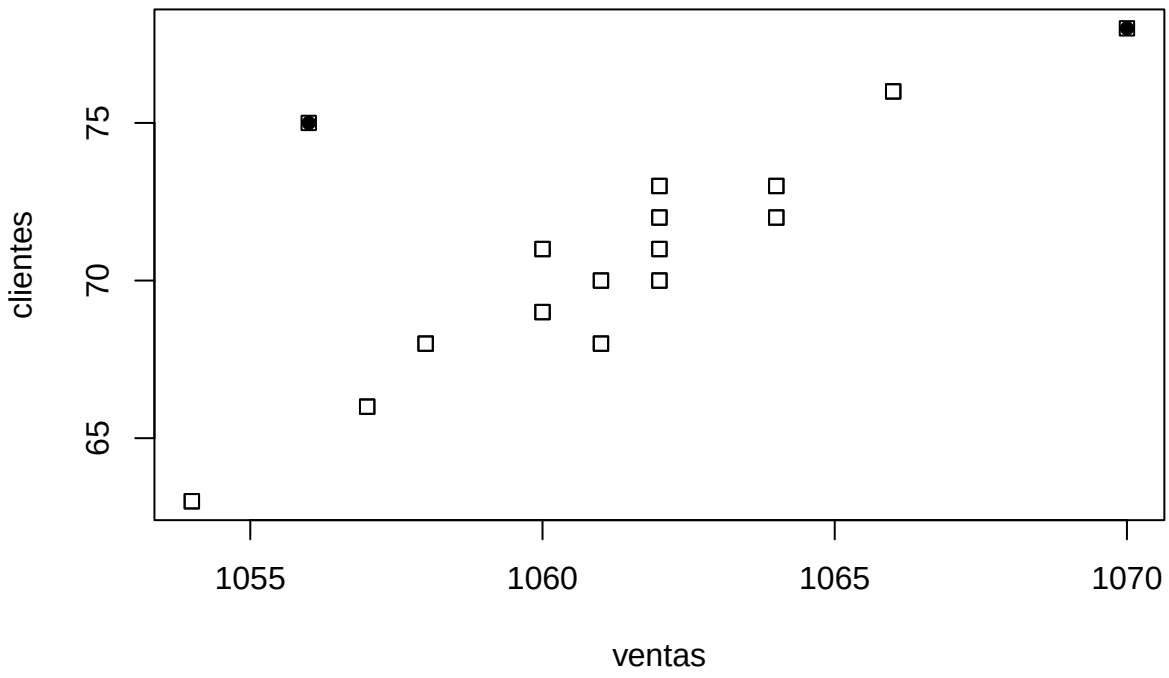
```
outlier2 <- rep(FALSE , nrow(datos))
outlier2[mah.ordenacion[1:num.outliers]] <- TRUE
```

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)
```



Despues de indicarle que punto queremos resaltar de las distancia slos gráfuicamos y lo podemos ver los autliers y el dato 16.

Ejercicio 2

Paquetrias necesarias

```
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
```

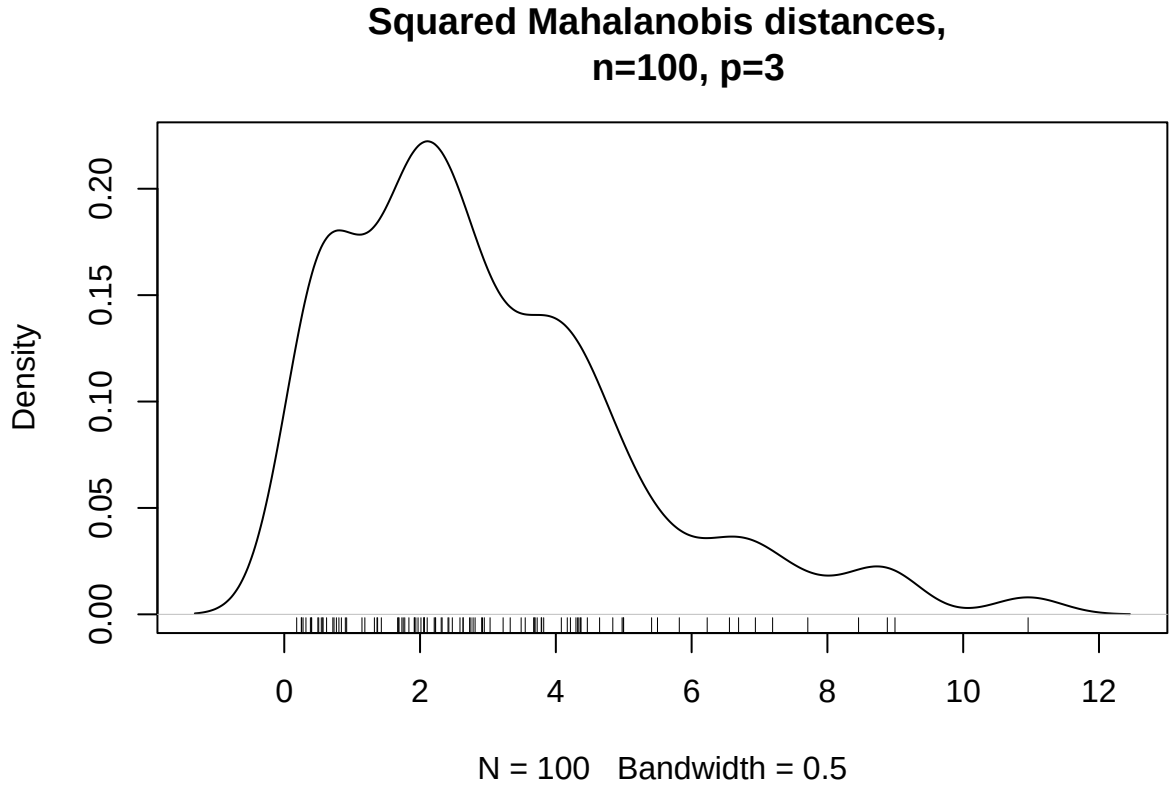
Se crea un vector y la varianza del mismo vector, cálculando la diatancia de mahalanobis aprtir de la varianza del primer objeto (ma).

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

Creamos uan matriz con **rnorm** con tres columnas despues se le indica que lo rsultante de "mahalanobis" lo coloque en la diagonal de la nueva matriz creadasi es igual a la suma de la multiplicación de **x*x**

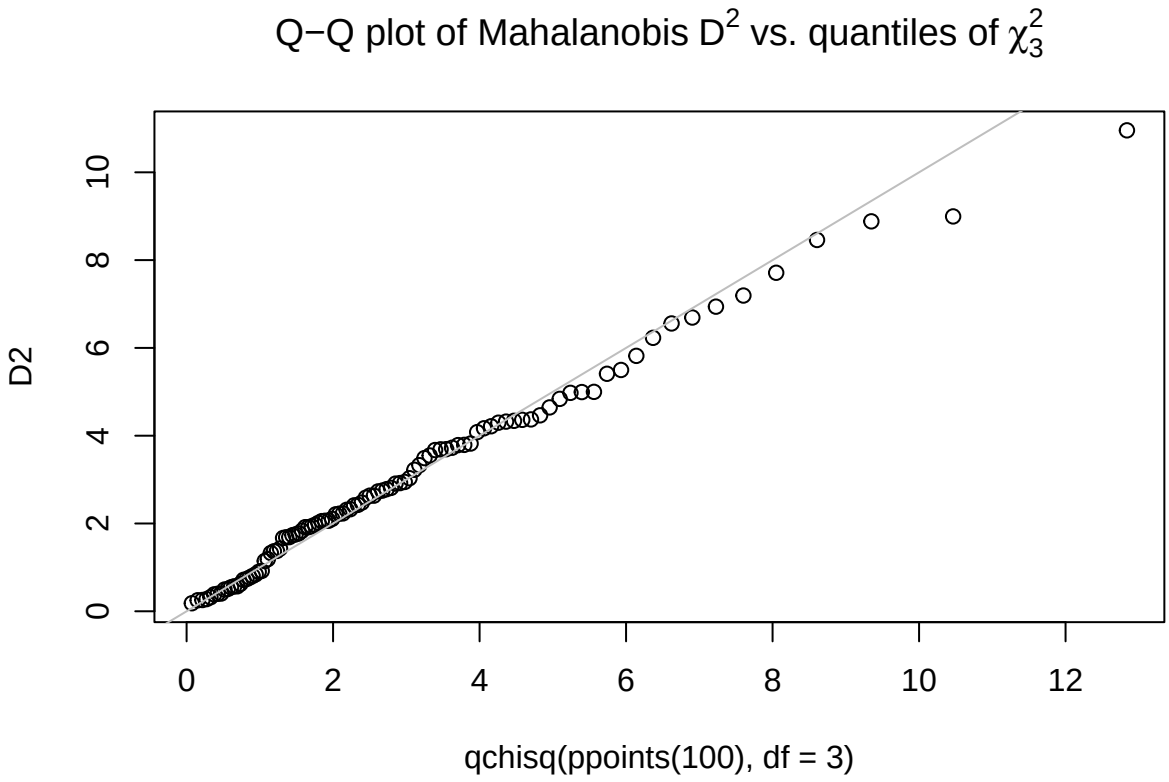
Here, D^2 = usual squared Euclidean distances

```
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)
```



La gráfica muetra las distancias

```
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')
```



Este gráfico muestra la distancias al cuadrado contra los cuantiles de la diatribución chi cuadrada.

Ejercicio 3

Diseñar un ejercicio utilizando la distancia de Mahalanobis.

Incluye:

1.- Planteamiento del problema.

Para calcular la distancia de mahalanobis debemos usar variables numéricas, que no necesaria mente estén en la misma escala de mediad ni midan lo mismo pues es lo bueno que se puede tener con este método de cálculo de distancia.

Usaremos la base *mtautos* que se encuentra dentro de la paquetería "**datos**" para R que tiene 11 variables y 32 observaciones que son distintas marcas de autos las variables son 11 y van desde la cilindrada del auto así como los caballos de fuerza, velocidad, etc.

2.- Simular los datos o utilizar una matriz Precargada en R.

Base y exploración

Librería necesaria

```
library(datos)
```

```
C<-data.frame(mtautos[3:4])
C
```

```
##                     cilindrada caballos
## Mazda RX4                160.0      110
## Mazda RX4 Wag            160.0      110
## Datsun 710               108.0       93
## Hornet 4 Drive           258.0      110
## Hornet Sportabout        360.0      175
## Valiant                  225.0      105
## Duster 360               360.0      245
## Merc 240D                146.7       62
## Merc 230                 140.8       95
## Merc 280                 167.6      123
## Merc 280C                167.6      123
## Merc 450SE               275.8      180
## Merc 450SL               275.8      180
## Merc 450SLC              275.8      180
## Cadillac Fleetwood       472.0      205
## Lincoln Continental      460.0      215
## Chrysler Imperial        440.0      230
## Fiat 128                  78.7       66
## Honda Civic               75.7       52
## Toyota Corolla            71.1       65
## Toyota Corona            120.1       97
## Dodge Challenger         318.0      150
## AMC Javelin              304.0      150
## Camaro Z28               350.0      245
## Pontiac Firebird         400.0      175
## Fiat X1-9                 79.0       66
## Porsche 914-2            120.3       91
## Lotus Europa              95.1      113
## Ford Pantera L           351.0      264
## Ferrari Dino             145.0      175
## Maserati Bora            301.0      335
## Volvo 142E               121.0      109
```

De la base original solo usamos 2 columnas aunque las demás columnas son numéricas, solo escogemos dos.

```
dim(C)
```

```
## [1] 32  2
```

```
anyNA(C)
```

```
## [1] FALSE
```

Exploramos la base en búsqueda de datos faltantes y visualizamos al dimensión.

Determinar el número de outlier que queremos encontrar.

```
numero.outliers <- 4
```

Como es un estudio con autlier determinamos cuantos serán y a partir de aquí se calculara la distancia

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

```
maha.ordenacion <- order(mahalanobis(C, colMeans(C), cov(C)), decreasing=TRUE)
maha.ordenacion
```

```
##  [1] 31 15 16 29 25 30 17 24  7 19  8 20 18 26 28  5  4 22  3  6 32 27 23 21  9
## [26]  1  2 10 11 12 13 14
```

Ordenamos las distancias de mahalanobis de los dato, las medias de las columnas y la covarianza de los datos y ordenados de mayor a menor; lo visualizamos para observar los datos, donde observamos que los datos 31,15,16 y 29 las distancias de mahalanobis son mayor y en los datos 11,12,13 y 14 las distancias son menores.

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

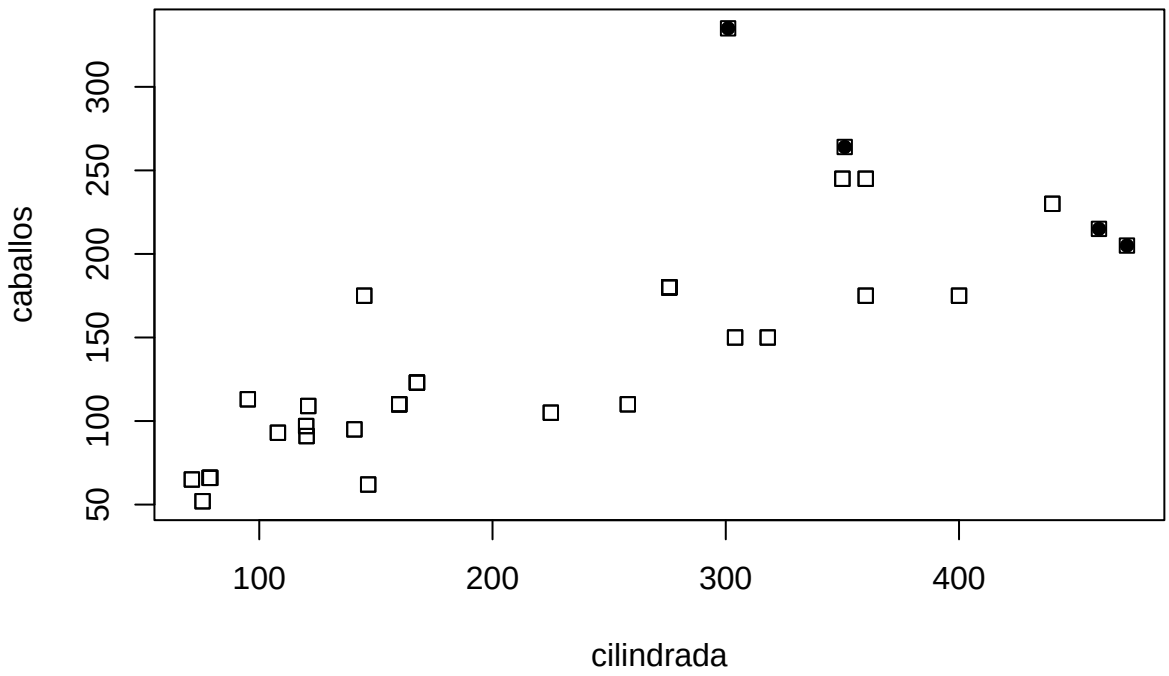
```
Outlier2 <- rep(FALSE , nrow(C))
Outlier2[maha.ordenacion[1:numero.outliers]] <- TRUE
```

Resaltar con un punto relleno los 2 valores outliers.

```
Colorear.outlier <- Outlier2 *16
```

Visualizar el gráfico con los datos destacando sus outlier.

```
plot(C , pch=0)
points(C, pch=Colorear.outlier)
```



3.- Dar tu interpretacion.

Cuando usamos solo dos variables y les calculamos la distancia podemos ver en nuestro gráfico qué tenemos tres outlires que resaltan, traducción son cuatro datos de las distancias calculadas que se encuentran más lejos de los demás, y eso serian los que en el ordenamiento vimos la distancia de los datos:31,15,16 y 29 que en el gráfico anterior están resaltados.