Distancia de Mahalanobis

2022-05-19

Cargaremos los datos con los cuales vamos a trabajar.

```
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)
```

Exploración de los datos

```
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 las distancias

El metodo de distancia de Mahalanobis mejora el metodo clásico de la distancia de Gauss elmiminando ele fecto que pueden producir la correlacion entre las variables a analisis.

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.

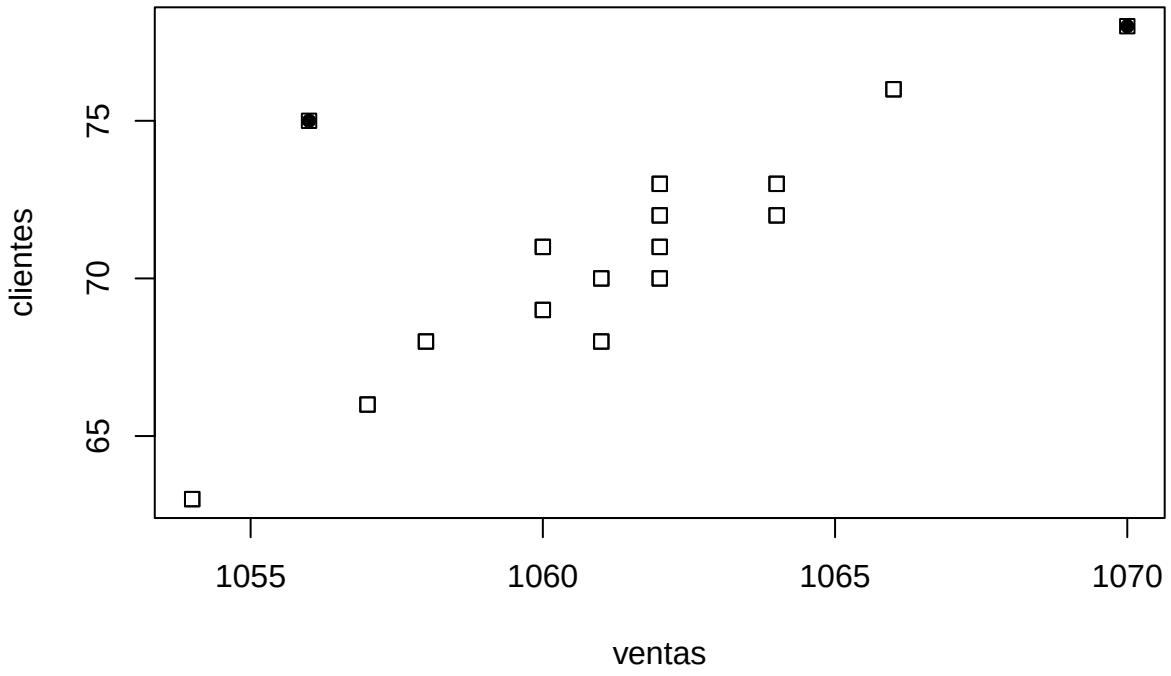
```
outlier2 <- rep(FALSE , nrow(datos))
outlier2[mah.ordenacion[1:num.outliers]] <- TRUE
```

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 EXTRA

Su utilidad radica en que es una forma de determinar la similitud entre dos variables aleatorias multidimensionales.

Para este ejercicio de la distancia de mahalanobis se trabaja en un a base de datos precargada en el paquete llamado *datos*. Ya instalado dicho paquete, llamaremos la libreria.

```
library(datos)
```

Para el siguiente paso se convertira en data frame la base de datos seleccionada la cual será *fiel*. Y la guardaremo en el objeto llamado *Z*.

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

Ya teniendo la base de datos para trabajar, se debe explorar para tener un mejor panorama del trabajo a realizar y que dicha base no tenga algo que pueda entorpecer el trabajo.

```
dim(Z)
```

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

```
anyNA(Z)
```

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

```
str(Z)
```

```
## '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 ...
```

Teniendo el conocimiento que no tenemos valores perdidos y que nuestra base tiene 272 filas y 2 columnas, la cual almacena 272 observaciones. Ahora es necesario conocer unas estadísticas básicas con la funcion Summary.

```
summary(Z)
```

```
##    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
```

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(Z , colMeans(Z), cov(Z)), decreasing=TRUE)
```

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

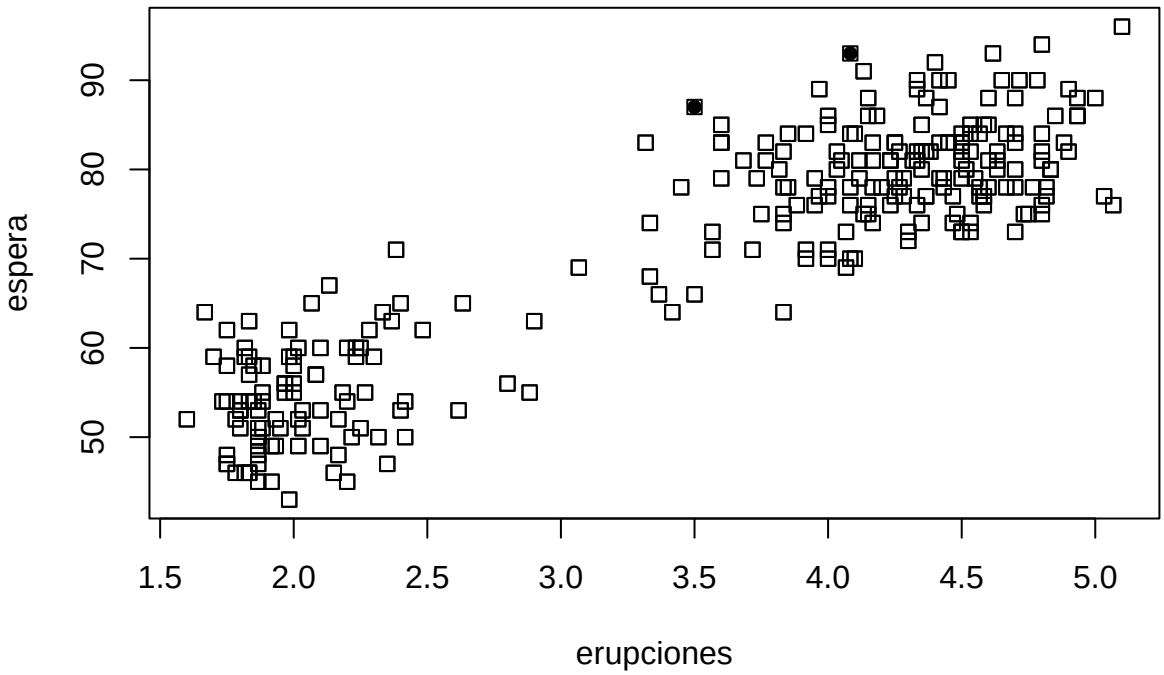
```
outlier2 <- rep(FALSE , nrow(Z))
outlier2[mah.ordenacion[1:num.outliers]] <- TRUE
```

Resaltar con un punto relleno los 2 valores outliers.

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

Visualizar el gráfico con los datos destacando sus outlier.

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


espera
50
60
70
80
90
1.5
2.0
2.5
3.0
3.5
4.0
4.5
5.0
erupciones