Cálculo de la distancia de Mahalanobis_Práctica de 3 momentos

Medel Colorado Yoselin Merari

2022-05-25

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.

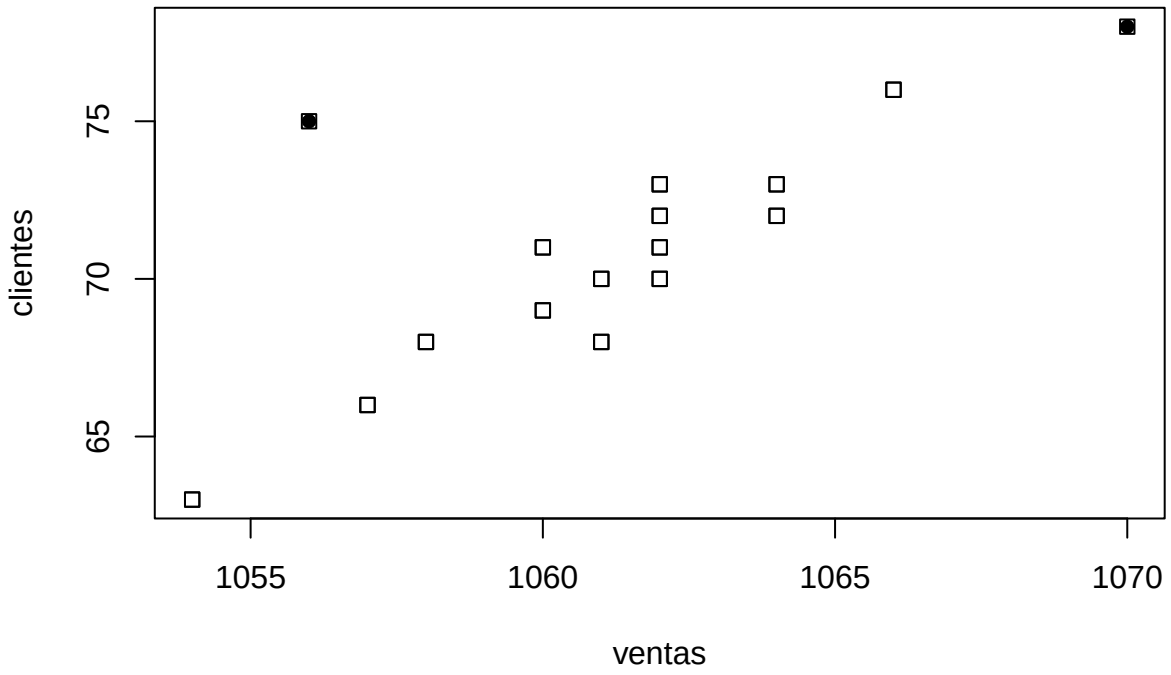
```
outlier2 <- rep(FALSE , nrow(datos))
outlier2[mah.ordenacion[1:num.outliers]] <- TRUE
```

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 (precargado en R) #————————————————-

```
require(graphics)

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

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

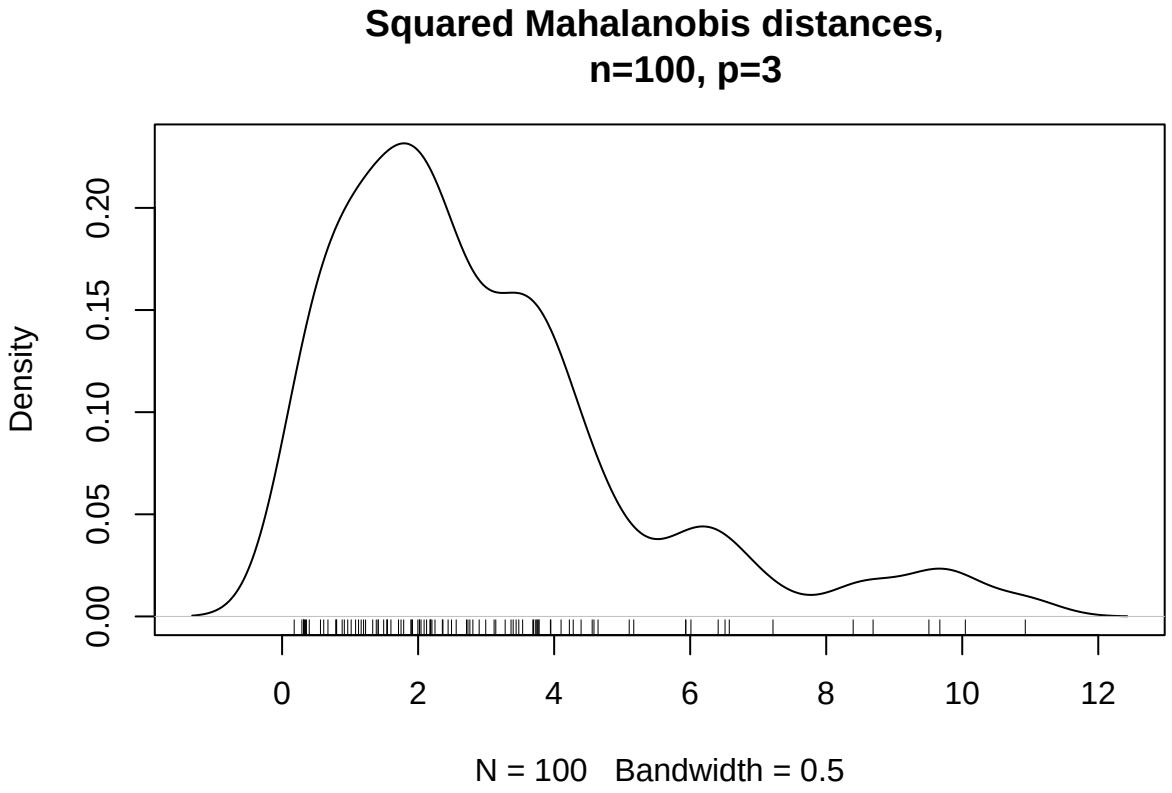
```
mahalanobis(c(0, 0), 1:2, S)
```

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

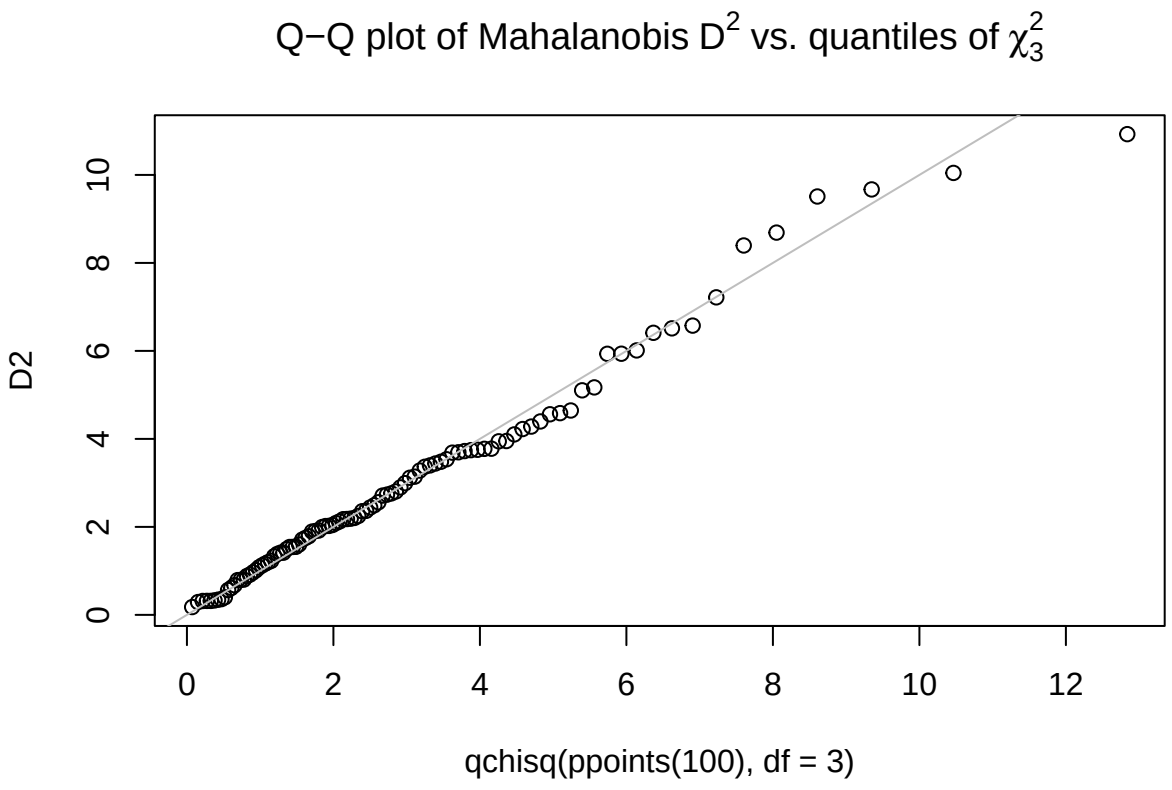
```
x <- matrix(rnorm(100*3), ncol = 3)
stopifnot(mahalanobis(x, 0,
                      diag(ncol(x))) == rowSums(x*x))
```

##- 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.

Nota: Una vez que terminaste subes el script a tu repositorio en GitHub. Sí te sobra tiempo puedes ir creando el pdf en markdown.

Instalar paquete

```
install.packages("datos")
library(datos)
```

Se hace una data.frame

```
Datos<- data.frame(datos :: fiel)
dim(fiel)
```

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

```
str(fiel)
```

```
## '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(fiel)
```

```
##    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
```

Cálculo de distancia

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(fiel , colMeans(fiel), cov(fiel)), decreasing=TRUE)
mah.ordenacion
```

```
##   [1] 158 197  58  76 265  46 161 203  17 211 160 151 242  95   8 249 269  70
##  [19]  66 127  51 131  69 115 170 267 149 218 111 193 135 188  65 271  89 206
##  [37]  92 178  26  80  47 144  44  75 106  45 255 119  14  22  39 270 117 177
##  [55] 254 235 134 103  37  90  63  94   6  19 234 148  25 263 121 209 171 213
##  [73] 208 192 261  42  55 184 199 223  93 221  77 179 272 130 102  21 146 108
##  [91]   2  12  38  54  99 166 150 159 137  50 162  96  52 185   9 122  68  11
## [109]  53 237  40 233 181 204 217 100 224 169 133  16 236 163 200  36 201 240
## [127]  83 110 153 173 182  72 124 231 113  27 168 191 139  31  59 120  48 259
## [145]  56 232 125 219  86 246 250 205  23   7 138 243   4  64 187 212 172 247
## [163] 251  91 194 126  78 167 109 190 142   1 156 129  18 266 147  32  61  10
## [181] 230  15 116 107 112 215 118 245   5  97  49 154  74  71 132 229  73 145
## [199] 256  43 186 262  62  88 207 198 140 183 264 101  84 180 143 104 210  30
## [217] 157 252 258 189 128 114  85 165 257 105 222 136 248   3 202  82 216 238
## [235] 268 241 175 176  41 123  60 141 196  81  34 228 260 220 239  20 164  13
## [253]  29  33 226  67 244  24  28 227 225 152 195  57  87  79  35  98 214 174
## [271] 253 155
```

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

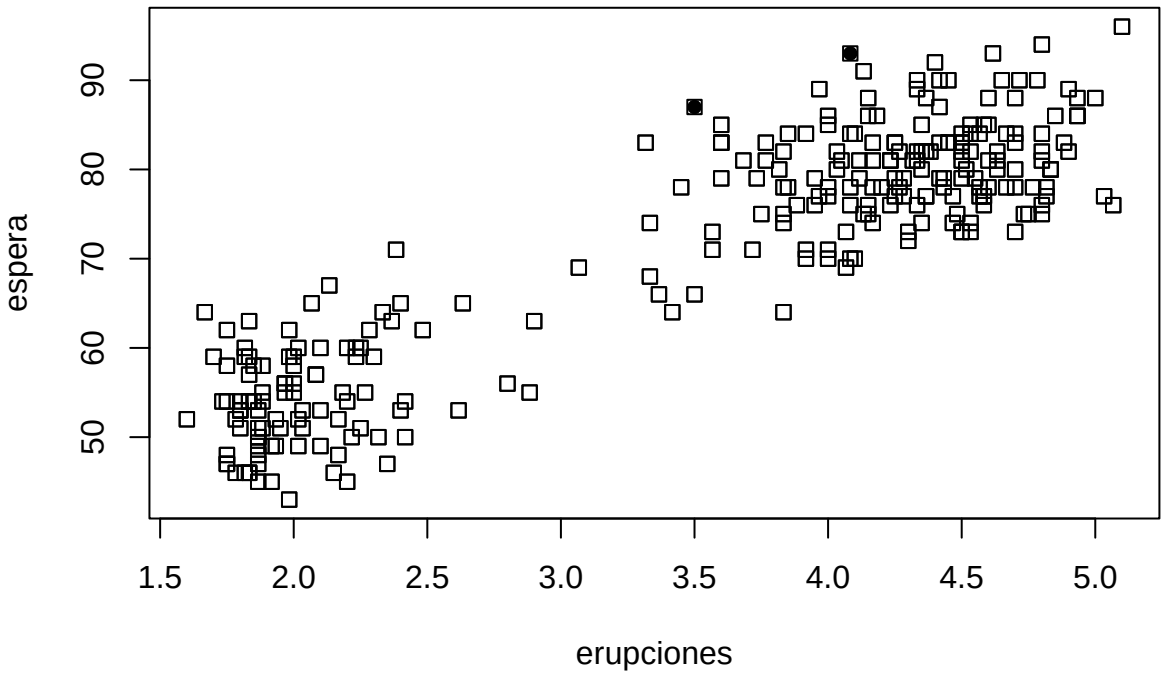
```
outlier2 <- rep(FALSE , nrow(fiel))
outlier2[mah.ordenacion[1:num.outliers]] <- TRUE
```

Resaltar con un punto relleno los 2 valores outliers

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

Visualizar el gráfico con los datos destacando sus outlier

```
plot(fiel , pch=0)
points(fiel, 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