

PECL5 - Fundamentos de la ciencia de datos

Mario Adán Herrero Alberto González Martínez
Branimir Stefanov Yanev Diego Gutiérrez Marco

12 de enero de 2021

Resumen

En el siguiente documento se presentan los resultados y la solución de la PECL5 del laboratorio de Fundamentos de la Ciencia de los Datos. En esta práctica utilizaremos R para realizar análisis de detección de datos anómalos de una muestra utilizando los distintos métodos estudiados en teoría.

Índice

1. Ejercicio 1 - Análisis de detección de datos anómalos	3
1.1. Introducción	3
1.2. Apartado 1 - Detección de datos anómalos. Medidas de ordenación	3
1.3. Apartado 2 - Detección de datos anómalos. Medidas de dispersión	5
1.4. Apartado 3 - Detección de datos anómalos. Regresión	6
1.5. Apartado 4 - Detección de datos anómalos. Algoritmo K-vecinos	7
2. Ejercicio 2 - Análisis de detección de datos anómalos	8
2.1. Introducción	8
2.2. Apartado 1 - Detección de datos anómalos. Medidas de ordenación	8
2.3. Apartado 2 - Detección de datos anómalos. Medidas de dispersión	10
2.4. Apartado 3 - Detección de datos anómalos. Regresión	11
2.5. Apartado 4 - Detección de datos anómalos. Algoritmo K-vecinos	12
3. Ejercicio 3- Análisis de detección de datos anómalos. LOF	13
4. Conclusiones	16

1. Ejercicio 1 - Análisis de detección de datos anómalos

1.1. Introducción

Este primer ejercicio consiste en la realización de cuatro apartados distintos en los que se realizará un análisis de detección de datos anómalos con R. Para la realización de estos apartados, se nos proporcionan dos muestras distintas, una primera que contiene valores de resistencia y densidad de distintos tipos de hormigón y una segunda, que está formada por calificaciones de estudiantes.

1. Análisis sobre medidas de ordenación (resistencia - muestra1).
2. Análisis sobre medidas de dispersión (densidad - muestra1).
3. Análisis sobre regresión de las variables (densidad/resistencia - muestra1)
4. Análisis mediante algoritmo K-vecinos (muestra2)

1.2. Apartado 1 - Detección de datos anómalos. Medidas de ordenación

En este primer apartado se realiza un análisis de detección de datos anómalos utilizando medidas de ordenación, método de caja y bigotes, sobre el valor de resistencia, para los diferentes tipos de hormigón que aparecen en la muestra1.

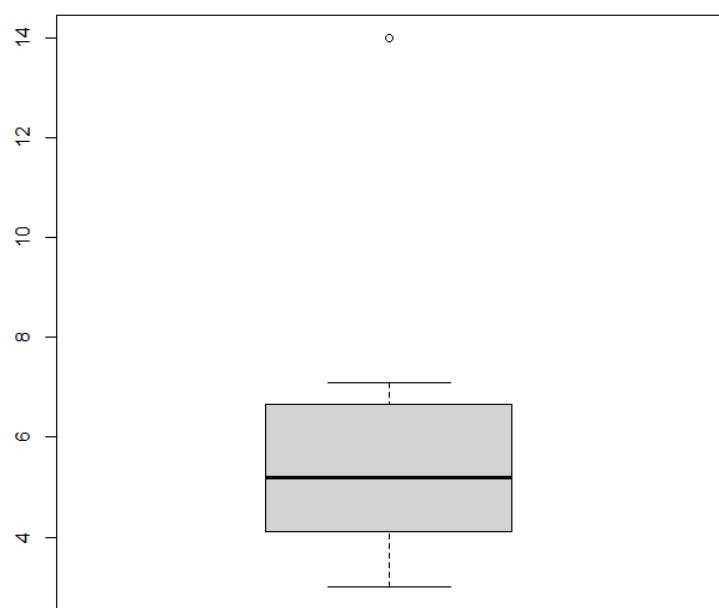
Cargamos primeramente los datos de la muestra1 que serán necesarios para la realización del primer apartado:

```
> #Cargamos los datos
> muestra1 <- t(matrix(c(3, 2, 3.5, 12, 4.7, 4.1, 5.2, 4.9, 7.1,
+                         6.1, 6.2, 5.2, 14, 5.3), 2, 7,
+                         dimnames = list(c("r", "d"))))
> muestra1 = data.frame(muestra1)
> muestra1

      r     d
1 3.0  2.0
2 3.5 12.0
3 4.7  4.1
4 5.2  4.9
5 7.1  6.1
6 6.2  5.2
7 14.0 5.3
```

Para mostrar los datos de los outliers de una muestra por pantalla utilizando el método de caja y bigotes utilizaremos la función de R *boxplot* que además de mostrarnos el método caja y bigotes tiene una opción, la cual se puede desactivar, para mostrar un plot.

```
> #Utilizamos la función quitando la generación del plot y mostramos
> boxplot(muestra1$r, range=1.5, plot = TRUE)
```



Gracias a *boxplot*, podemos visualizar de forma gráfica los datos.

La función *boxplot*, como podemos ver, nos muestra la información del análisis de los datos anómalos, por ello a continuación, implementaremos de otro modo el método de caja bigotes.

Primero hallaremos los cuartiles (q1, q3) de la muestra1, para poder luego poder calcular el intervalo de datos que consideramos normales, es decir, el intervalo de datos no anómalos

```
> #Calculamos los cuartiles
> q1 <- quantile(muestra1$r, 0.25)
> q3 <- quantile(muestra1$r, 0.75)
> #Calculamos el intervalo de valores normales
> s = 1.5
> intervalo <- c(q1 - s * (q3 - q1), q1 + s*(q3-q1))
> intervalo

25%    25%
0.275 7.925
```

Una vez realizado lo anterior y teniendo el intervalo de valores normales, recorreremos todos los valores de la muestra y comprobaremos cuáles de esos valores son datos anómalos y cuáles no lo son, es decir, cuales están dentro, o no, del intervalo.

```
> for (i in 1:length(muestra1$r))
+ {
+   if(muestra1$r[i] < intervalo[1] || muestra1$r[i] > intervalo[2])
+   {
+     cat("DATO - [", muestra1$r[i], "] es anómalo.\n")
+   }
+ }
```

DATO - [14] es anómalo.

Como vemos, a diferencia del *boxplot*, con este algoritmo podemos obtener todos los valores anómalos como una variable para, en caso de necesitarlo, poder trabajar con ellos.

1.3. Apartado 2 - Detección de datos anómalos. Medidas de dispersión

En este apartado realizamos un análisis de detección de datos anómalos utilizando medidas de dispersión sobre la densidad, desviación típica, de la muestra1.

Usamos la variable muestra1 creada en el apartado anterior y usamos la desviación típica para calcular los intervalos de datos normales.

```
> intervalo <- c(mean(muestra1$d) - 2*sd(muestra1$d), mean(muestra1$d) + 2*sd(muestra1$d))
> intervalo

[1] -0.5146825 11.8289682

> sdd <- sqrt(var(muestra1$d) * (length(muestra1$d)-1 / length(muestra1$d)))
```

Se comprueba que valores se encuentran dentro del intervalo especificado para determinar cuales son datos anómalos.

```
> for(i in 1:length(muestra1$d)){
+   if(muestra1$d[i] < intervalo[1] || muestra1$d[i] > intervalo[2]) {
+     cat("DATO - [", muestra1$d[i], "] es anómalo.\n")
+   }
+ }

DATO - [ 12 ] es anómalo.
```

1.4. Apartado 3 - Detección de datos anómalos. Regresión

Para el tercer apartado realizaremos un análisis de detección de datos anómalos sobre la regresión de las variables densidad en función de resistencia, utilizando, para ello, el error estándar de los residuos sobre la muestra1.

Cómo utilizamos la muestra1, y esta ya está definida en el primer apartado, no hace falta volver a definirla de nuevo. Sobre esta muestra, calculamos la regresión y extraemos los residuos.

```
> #Calculamos la regresión
> regresion = lm(muestra1$d~muestra1$r)
> regresion

Call:
lm(formula = muestra1$d ~ muestra1$r)

Coefficients:
(Intercept) muestra1$r
6.01445      -0.05723

> #Calculamos el residuo
> residuos = summary(regresion)$residuals
> residuos

 1          2          3          4          5          6          7 
-3.8427477  6.1858698 -1.6454482 -0.8168308  0.4919157 -0.4595958  0.0868370
```

A continuación, calculamos el error estándar en función de la densidad y los residuos calculados anteriormente:

```
> #Calculamos el error
> error = sqrt(sum(residuos**2)/length(muestra1$d))
> error

[1] 2.850242
```

Finalmente, identificamos como anómalos los datos cuyo valor absoluto supere el rango correspondiente al grado de outlier $d = 1,5$. Finalmente, para obtener los datos anómalos, identificamos qué datos tienen un valor absoluto superior al rango que corresponde al grado de outlier.

```

> grado_outlier = 1.5
> dsr = grado_outlier * error
> for (i in 1:length(muestra1$r))
+ {
+   if(abs(residuos[i]) > dsr)
+   {
+     cat("DATO - [", muestra1$d[i], "] es anómalo.\n")
+   }
+ }

DATO - [ 12 ] es anómalo.

```

1.5. Apartado 4 - Detección de datos anómalos. Algoritmo K-vecinos

En este apartado utilizaremos el algoritmo K-vecinos para la detección de los datos anómalos de la muestra de datos de las calificaciones.

Utilizaremos un valor de $K = 4$ y un grado de outlier $d = 2.5$.

Primero, cargamos nuestros datos en la variable *calificaciones*.

```

> #Cargamos las calificaciones
> calificaciones <- matrix(c(4, 4, 4, 3, 5, 5, 1, 1, 5, 4), 2,5)
> calificaciones <- t(calificaciones )
> calificaciones

 [,1] [,2]
[1,]    4    4
[2,]    4    3
[3,]    5    5
[4,]    1    1
[5,]    5    4

```

A continuación, crearemos una matriz con las distancias entre los puntos de la muestra y la ordenaremos de forma ascendente.

```

> #Distancias sin ordenar
> distancias <- as.matrix(dist(calificaciones ))
> distancias <- matrix(distancias, 5, 5)
> distancias

 [,1]      [,2]      [,3]      [,4]      [,5]
[1,] 0.000000 1.000000 1.414214 4.242641 1.000000
[2,] 1.000000 0.000000 2.236068 3.605551 1.414214
[3,] 1.414214 2.236068 0.000000 5.656854 1.000000
[4,] 4.242641 3.605551 5.656854 0.000000 5.000000
[5,] 1.000000 1.414214 1.000000 5.000000 0.000000

> for(i in 1:5){
+   distancias[,i] = sort(distancias[,i])
+ }
> #Distancias ordenadas
> distanciasOrdenadas <- distancias
> distanciasOrdenadas

```

```

[,1]      [,2]      [,3]      [,4]      [,5]
[1,] 0.000000 0.000000 0.000000 0.000000 0.000000
[2,] 1.000000 1.000000 1.000000 3.605551 1.000000
[3,] 1.000000 1.414214 1.414214 4.242641 1.000000
[4,] 1.414214 2.236068 2.236068 5.000000 1.414214
[5,] 4.242641 3.605551 5.656854 5.656854 5.000000

```

Una vez hecho esto, ya podemos saber qué datos son anómalos: Consideraremos un dato anómalo cualquier dato su vecino K esté a una distancia mayor que el grado de outlier d. Cualquier valor cuyo vecino K = 4 esté a una distancia mayor que d = 2.5 será considerado anómalo.

```

> for(i in 1:5){
+   if(distanciasOrdenadas[4,i] > 2.5) {
+     cat("[", i, "] es un outlier\n")
+   }
+ }

[ 4 ] es un outlier

```

2. Ejercicio 2 - Análisis de detección de datos anómalos

2.1. Introducción

Para el segundo ejercicio, realizaremos el mismo análisis de los datos realizados en el apartado 1, pero utilizando una muestra diferente. Esta muestra tiene los datos sobre el peso y la altura de estudiantes de estadística. Los análisis que se realizarán en este apartado serán los siguientes:

1. Análisis sobre medidas de ordenación
2. Análisis sobre medidas de dispersión
3. Análisis sobre regresión de las variables
4. Análisis mediante algoritmo K-vecinos

2.2. Apartado 1 - Detección de datos anómalos. Medidas de ordenación

En este primer apartado se realiza un análisis de detección de datos anómalos utilizando medidas de ordenación, método de caja y bigotes, sobre el valor de la altura, para los diferentes estudiante de estadística que aparecen en el conjunto alumnos.

Leemos primeramente los datos de la muestra de alumnos que serán necesarios para la realización del primer apartado:

```

> #Leemos los datos del csv
>
> library("xlsx")
> alumnos <- read.xlsx("./data/datos_altura_peso.xlsx", 1)
> alumnos

```

```

altura peso
1     181   73
2     179   75
3     182   78
4     177   70
5     175   69
6     180   79
7     215  100
8     156   50

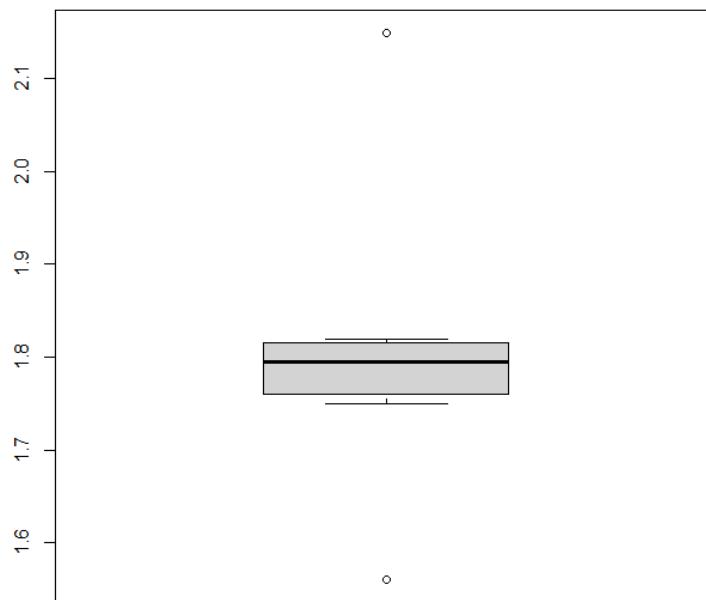
```

Para mostrar los datos de los outliers de una muestra por pantalla utilizando el método de caja y bigotes utilizaremos la función de R *boxplot* que además de mostrarnos el método caja y bigotes tiene una opción, la cual se puede desactivar, para mostrar un plot.

```

> #Utilizamos la función quitando la generación del plot y mostramos
> boxplot(alumnos$altura, range=1.5, plot = TRUE)

```



Gracias a *boxplot*, podemos visualizar de forma gráfica los datos.

La función *boxplot*, como podemos ver, nos muestra la información del análisis de los datos anómalos, por ello a continuación, implementaremos de otro modo el método de caja bigotes.

Primero hallaremos los cuartiles (q_1 , q_3) de las alturas de los alumnos, para poder luego poder calcular el intervalo de datos que consideramos normales, es decir, el intervalo de datos no anómalos

```
> #Calculamos los cuartiles
> q1 <- quantile(alumnos$altura, 0.25)
> q3 <- quantile(alumnos$altura, 0.75)
> #Calculamos el intervalo de valores normales
> s = 1.5
> intervalo <- c(q1 - s * (q3 - q1), q1 + s*(q3-q1))
> intervalo

25%      25%
169.375 183.625
```

Una vez realizado lo anterior y teniendo el intervalo de valores normales, recorreremos todos los valores de la muestra y comprobaremos cuáles de esos valores son datos anómalos y cuáles no lo son, es decir, cuales están dentro, o no, del intervalo.

```
> for (i in 1:length(alumnos$altura))
+ {
+   if(alumnos$altura[i] < intervalo[1] || alumnos$altura[i] > intervalo[2])
+   {
+     cat("DATO - [", alumnos$altura[i], "] es anómalo.\n")
+   }
+ }
```

DATO - [215] es anómalo.
DATO - [156] es anómalo.

Como vemos, a diferencia del *boxplot*, con este algoritmo podemos obtener todos los valores anómalos como una variable para, en caso de necesitarlo, poder trabajar con ellos.

2.3. Apartado 2 - Detección de datos anómalos. Medidas de dispersión

En este apartado realizamos un análisis de detección de datos anómalos utilizando medidas de dispersión sobre el peso, desviación típica, de la muestra de alumnos.

Usamos la variable alumnos creada en el apartado anterior y usamos la desviación típica para calcular los intervalos de datos normales.

```
> intervalo <- c(mean(alumnos$peso) - 2*sd(alumnos$peso), mean(alumnos$peso) + 2*sd(alumnos$peso))
> intervalo

[1] 46.62496 101.87504
```

```

> sdd <- sqrt(var(alumnos$peso) * (length(alumnos$peso)-1 / length(alumnos$peso)))
> sdd
[1] 38.76129

```

Se comprueba que valores se encuentran dentro del intervalo especificado para determinar cuales son datos anómalos.

```

> for(i in 1:length(alumnos$peso)){
+   if(alumnos$peso[i] < intervalo[1] || alumnos$peso[i] > intervalo[2]) {
+     cat("DATO - [",alumnos$peso[i], "] es anómalo.\n")
+   }
+ }

```

2.4. Apartado 3 - Detección de datos anómalos. Regresión

Para el tercer apartado realizaremos un análisis de detección de datos anómalos sobre la regresión de las variables peso en función de altura, utilizando, para ello, el error estándar de los residuos sobre la muestra de alumnos.

```

> #Calculamos la regresión
> regresion = lm(alumnos$peso~alumnos$altura)
> regresion

Call:
lm(formula = alumnos$peso ~ alumnos$altura)

Coefficients:
(Intercept)  alumnos$altura
-75.8961          0.8313

> #Calculamos el residuo
> residuos = summary(regresion)$residuals
> residuos

 1         2         3         4         5         6         7
-1.5617221  2.1007958  2.6070190 -1.2366864 -0.5741685  5.2695368 -2.8245256
 8
-3.7802489

```

A continuación, calculamos el error estándar en función de la peso y los residuos calculados anteriormente:

```

> #Calculamos el error
> error = sqrt(sum(residuos**2)/length(alumnos$peso))
> error
[1] 2.862346

```

Finalmente, identificamos como anómalos los datos cuyo valor absoluto supere el rango correspondiente al grado de outlier $d = 1,5$. Finalmente, para obtener los datos anómalos, identificamos qué datos tienen un valor absoluto superior al rango que corresponde al grado de outlier.

```

> grado_outlier = 1.5
> dsr = grado_outlier * error
> for (i in 1:length(alumnos$altura))
+ {
+   if(abs(residuos[i]) > dsr)
+   {
+     cat("DATO - [", alumnos$peso[i], "] es anómalo.\n")
+   }
+ }

DATO - [ 79 ] es anómalo.

```

2.5. Apartado 4 - Detección de datos anómalos. Algoritmo K-vecinos

En este apartado utilizaremos el algoritmo K-vecinos para la detección de los datos anómalos de la muestra de datos de las alturas y pesos de alumnos.

Utilizaremos un valor de K = 4 y un grado de outlier d = 2.5.

Utilizaremos la muestra que tenemos cargada en la variable /textitalumnos. A continuación, crearemos una matriz con las distancias entre los puntos de la muestra y la ordenaremos de forma ascendente.

```

> #Distancias sin ordenar
> distancias <- as.matrix(dist(alumnos ))
> distancias <- matrix(distancias, 8, 8)
> distancias

[,1]      [,2]      [,3]      [,4]      [,5]      [,6]      [,7]
[1,] 0.000000 2.828427 5.099020 5.000000 7.211103 6.082763 43.41659
[2,] 2.828427 0.000000 4.242641 5.385165 7.211103 4.123106 43.82921
[3,] 5.099020 4.242641 0.000000 9.433981 11.401754 2.236068 39.66106
[4,] 5.000000 5.385165 9.433981 0.000000 2.236068 9.486833 48.41487
[5,] 7.211103 7.211103 11.401754 2.236068 0.000000 11.180340 50.60632
[6,] 6.082763 4.123106 2.236068 9.486833 11.180340 0.000000 40.81666
[7,] 43.416587 43.829214 39.661064 48.414874 50.606324 40.816663 0.00000
[8,] 33.970576 33.970576 38.209946 29.000000 26.870058 37.643060 77.33693

[,8]
[1,] 33.97058
[2,] 33.97058
[3,] 38.20995
[4,] 29.00000
[5,] 26.87006
[6,] 37.64306
[7,] 77.33693
[8,] 0.00000

> for(i in 1:8){
+   distancias[,i] = sort(distancias[,i])
+ }
> #Distancias ordenadas
> distanciasOrdenadas <- distancias
> distanciasOrdenadas

```

	[,1]	[,2]	[,3]	[,4]	[,5]	[,6]	[,7]
[1,]	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
[2,]	2.828427	2.828427	2.236068	2.236068	2.236068	2.236068	39.66106
[3,]	5.000000	4.123106	4.242641	5.000000	7.211103	4.123106	40.81666
[4,]	5.099020	4.242641	5.099020	5.385165	7.211103	6.082763	43.41659
[5,]	6.082763	5.385165	9.433981	9.433981	11.180340	9.486833	43.82921
[6,]	7.211103	7.211103	11.401754	9.486833	11.401754	11.180340	48.41487
[7,]	33.970576	33.970576	38.209946	29.000000	26.870058	37.643060	50.60632
[8,]	43.416587	43.829214	39.661064	48.414874	50.606324	40.816663	77.33693
	[,8]						
[1,]	0.00000						
[2,]	26.87006						
[3,]	29.00000						
[4,]	33.97058						
[5,]	33.97058						
[6,]	37.64306						
[7,]	38.20995						
[8,]	77.33693						

Una vez hecho esto, ya podemos saber qué datos son anómalos: Consideraremos un dato anómalo cualquier dato su vecino K esté a una distancia mayor que el grado de outlier d. Cualquier valor cuyo vecino K = 4 esté a una distancia mayor que d = 2.5 será considerado anómalo.

```
> for(i in 1:8){
+   if(distanciasOrdenadas[4,i] > 2.5) {
+     cat("[", i, "] es un outlier\n")
+   }
+ }

[ 1 ] es un outlier
[ 2 ] es un outlier
[ 3 ] es un outlier
[ 4 ] es un outlier
[ 5 ] es un outlier
[ 6 ] es un outlier
[ 7 ] es un outlier
[ 8 ] es un outlier
```

3. Ejercicio 3- Análisis de detección de datos anómalos. LOF

En este ejercicio utilizaremos el algoritmo LOF (Local Outlier Factor) para determinar los outliers o datos anómalos del siguiente conjunto de datos.

Calificaciones - Teoría, Laboratorio

1. 4, 4
2. 4, 3
3. 5, 5

4. 1, 1

5. 5, 4

Para la resolución de este ejercicio haremos uso de funciones pertenecientes a un paquete externo llamado *dbscan*. Para ello instalamos el paquete.

```
> install.packages ("dbscan")
```

Utilizamos el comando `/textitlibrary` para cargar el paquete

```
> library(dbscan)
```

Cargamos los datos de la muestra de calificaciones

```
> calificaciones<- matrix(c(4, 4, 4, 3, 5, 5, 1, 1, 5, 4), 2,5)
> calificaciones<- t(calificaciones)
> calificaciones

 [,1] [,2]
[1,]    4    4
[2,]    4    3
[3,]    5    5
[4,]    1    1
[5,]    5    4
```

Calculamos el factor de outlier de cada valor de la muestra con la función `/textitlof` con $k=3$

```
> lof <- lof(calificaciones, k=3)
> lof
```

```
[1] 1.1081851 0.9069197 0.9069197 2.3007780 1.1081851
```

Resumen sobre los factores de outlier (mínimo, 1er cu, mediana, media, 3rd cu, max)

```
> summary(lof)
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
0.9069	0.9069	1.1082	1.2662	1.1082	2.3008

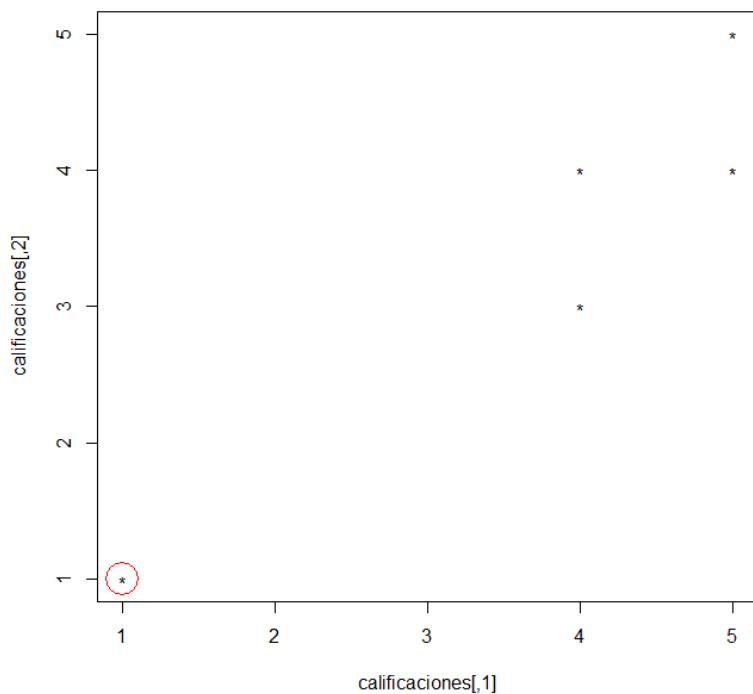
Representación gráfica de los factores de outlier de cada dato

```
> plot(calificaciones, pch = "*", main = "LOF (k=3)")
```

Marcamos con un círculo rojo los datos que consideramos anómalos/outliers

```
> points(calificaciones, cex = (lof-1)*3, pch = 1, col="red")
```

LOF (k=3)



Como vimos en teoría el punto 4 1,1 es considerado un outlier dentro de la muestra.

4. Conclusiones

Mediante esta práctica hemos empleado herramientas que permiten la realización de análisis de clasificación para un conjunto de datos mediante R. Se ha aprendido a utilizar el algoritmo de construcción de árboles de decisión de Hunt y a mostrar e interpretar este. También, se ha aprendido a realizar un análisis de regresión lineal e interpretar los resultados obtenidos.