



FUNDAMENTOS DE LA CIENCIA DE DATOS

---

## Prueba de Laboratorio 2 (PL2)

---

**Jorge Revenga Martín de Vidales**  
**Ángel Salgado Aldao**  
**Adrián García**

Grado en Ingeniería Informática  
Universidad de Alcalá

24 de noviembre de 2023

# Índice

<b>1. Introducción - Consideraciones previas</b>	<b>2</b>
1.1. Uso de RStudio . . . . .	2
1.2. Introducción de datos de Excel/CSV en R . . . . .	2
<b>2. Ejercicios con ayuda del profesor</b>	<b>4</b>
2.1. Análisis de clasificación no supervisada . . . . .	4
2.1.1. k-Means . . . . .	4
2.1.2. Clusterización Jerárquica Aglomerativa . . . . .	12
2.2. Análisis de clasificación supervisada . . . . .	13
2.2.1. Árboles de decisión . . . . .	13
2.2.2. Regresión . . . . .	13
<b>3. Ejercicios de forma autónoma</b>	<b>14</b>
3.1. Análisis de clasificación no supervisada . . . . .	14
3.1.1. K-means . . . . .	14
3.1.2. Clusterización Jerárquica Aglomerativa . . . . .	14
3.2. Análisis de clasificación supervisada . . . . .	16
3.2.1. Árboles de decisión . . . . .	16
3.2.2. Regresión . . . . .	16

# 1. Introducción - Consideraciones previas

## 1.1. Uso de RStudio

Para utilizar una función en R se escribe el nombre de la función, seguido de los parámetros de entrada entre paréntesis e.g.: `función(parámetros)`

- Función `contributors()`: Muestra los creadores del programa (R)
- Función `help()`: Abre un HTML con información sobre la función `help()` o de la función entre paréntesis de haberla. Para todas las funciones que programemos (para todas las que existan) en R debe poder usarse la función `help()`.

En el archivo HTML se distinguen varios elementos:

- función {paquete}: la función de la que se obtiene información seguida del paquete al que pertenece.
  - Description: descripción de la función.
  - Usage: aparece la función y todos los argumentos que se le pueden introducir.
  - Arguments: Explicación de los argumentos o parámetros.
  - Details: Detalles adicionales de la función.
  - Offline help: Ayuda sin conexión.
  - Note: Nota del autor.
  - References: Referencias.
  - Examples: Ejemplos de uso de la función.
- Función `getwd()` se utiliza para obtener el directorio de trabajo actual (working directory).
  - Función `setwd("C:/...")` permite cambiar el nuevo directorio de trabajo en el que queramos trabajar.
  - `help.start()`: Manda a un compendio de todas las ayudas disponibles para trabajar con R.
  - Función `list.files()`: Muestra todos los archivos en el directorio. `dir()` hace lo mismo.

## 1.2. Introducción de datos de Excel/CSV en R

Para poder leer archivos .xlsx y .csv seguimos un par de pasos:

- Archivos .csv:
  - Primero se crea el archivo .csv y se introducen los datos por filas (los datos de la fila i, se escriben en la celda (i,1)) y separando dichos datos con un delimitador (" , " ; ").

- Después se usa la función `read.csv(ruta del archivo, sep = ";")`, la cual es parte del paquete base de R, para leer los datos del archivo. Recibe como parámetros la ruta del archivo y el delimitador que usa para separar los datos.
- Archivos .xlsx:
- Primero se crea el archivo .xlsc y se introducen los datos en forma de matriz, escribiendo cada dato en una celda.
  - Después se carga la librería *openxlsx* y se usa la función `read_excel(ruta del archivo)`, introduciendo la ruta del archivo por parámetro.

## 2. Ejercicios con ayuda del profesor

Realización de cuatro ejercicios con ayuda del profesor en los que se van a realizar, utilizando el entorno R, dos análisis de clasificación no supervisada y dos análisis de clasificación supervisada, aplicando todos los conceptos teóricos vistos en cada lección.

### 2.1. Análisis de clasificación no supervisada

#### 2.1.1. k-Means

El primer conjunto de datos, que se empleará para realizar el análisis de clasificación no supervisada con k-Means, estará formado por las siguientes 8 calificaciones de estudiantes: 1.{4, 4}; 2.{3, 5}; 3.{1, 2}; 4.{5, 5}; 5.{0, 1}; 6.{2, 2}; 7.{4, 5}; 8.{2, 1}, donde las características de las calificaciones son: {Teoría, Laboratorio}.

**Solución:**

■ `m<-matrix(c(4,4, 3,5, 1,2, 5,5, 0,1, 2,2, 4,5, 2,1),2,8): Explicacion`

```
> m<-matrix(c(4,4, 3,5, 1,2, 5,5, 0,1, 2,2, 4,5, 2,1),2,8)
> (m<-t(m))
```

```
      [,1] [,2]
[1,]    4    4
[2,]    3    5
[3,]    1    2
[4,]    5    5
[5,]    0    1
[6,]    2    2
[7,]    4    5
[8,]    2    1
```

■ `c<-matrix(c(0,1,2,2),2,2): Explicacion`

```
> c<-matrix(c(0,1,2,2),2,2)
> (c<-t(c))
```

```
      [,1] [,2]
[1,]    0    1
[2,]    2    2
```

■ `(clasificacionns=(kmeans(m,c,4))): Explicacion`

```
> (clasificacionns=(kmeans(m,c,4)))
```

K-means clustering with 2 clusters of sizes 4, 4

Cluster means:

```
      [,1] [,2]
1 1.25 1.50
2 4.00 4.75
```

Clustering vector:

```
[1] 2 2 1 2 1 1 2 1
```

Within cluster sum of squares by cluster:

```
[1] 3.75 2.75
(between_SS / total_SS = 84.8 %)
```

Available components:

```
[1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
[6] "betweenss"    "size"         "iter"         "ifault"
```

■ : Explicacion

```
> (m=cbind(clasificacionns$cluster,m))
```

```
      [,1] [,2] [,3]
[1,]     2     4     4
[2,]     2     3     5
[3,]     1     1     2
[4,]     2     5     5
[5,]     1     0     1
[6,]     1     2     2
[7,]     2     4     5
[8,]     1     2     1
```

■ : Explicacion

```
> mc1=subset(m,m[,1]==1)
> mc2=subset(m,m[,1]==2)
> mc1
```

```
      [,1] [,2] [,3]
[1,]     1     1     2
[2,]     1     0     1
[3,]     1     2     2
[4,]     1     2     1
```

```
> mc2
```

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

■ : Explicacion

```
> (mc1=mc1[,-1])
```

```

      [,1] [,2]
[1,]    1    2
[2,]    0    1
[3,]    2    2
[4,]    2    1

```

```
> (mc2=mc2[, -1])
```

```

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

```

■ : Explicacion

```

> install.packages("LearnClust")
> library(LearnClust)
> search()

```

```

[1] ".GlobalEnv"          "package:LearnClust" "package:magick"      "package:clus
[5] "package:arules"       "package:Matrix"     "package:stats"      "package:grap
[9] "package:grDevices"    "package:utils"       "package:datasets"   "package:meth
[13] "Autoloads"           "package:base"

```

■ : Explicacion

```

> m<-matrix(c(0.89,2.94, 4.36,5.21, 3.75,1.12, 6.25,3.14, 4.1,1.8, 3.9,4.27),2,6)
> (m<-t(m))

```

```

      [,1] [,2]
[1,] 0.89 2.94
[2,] 4.36 5.21
[3,] 3.75 1.12
[4,] 6.25 3.14
[5,] 4.10 1.80
[6,] 3.90 4.27

```

■ : Explicacion

```
> agglomerativeHC(m, 'EUC', 'MIN')
```

```

$dendrogram
Number of objects: 6

```

```

$clusters
$clusters[[1]]
      X1    X2
1 0.89 2.94

```

```
$clusters[[2]]  
      X1  X2  
1 4.36 5.21
```

```
$clusters[[3]]  
      X1  X2  
1 3.75 1.12
```

```
$clusters[[4]]  
      X1  X2  
1 6.25 3.14
```

```
$clusters[[5]]  
      X1  X2  
1 4.1 1.8
```

```
$clusters[[6]]  
      X1  X2  
1 3.9 4.27
```

```
$clusters[[7]]  
      X1  X2  
1 3.75 1.12  
2 4.10 1.80
```

```
$clusters[[8]]  
      X1  X2  
1 4.36 5.21  
2 3.90 4.27
```

```
$clusters[[9]]  
      X1  X2  
1 3.75 1.12  
2 4.10 1.80  
3 4.36 5.21  
4 3.90 4.27
```

```
$clusters[[10]]  
      X1  X2  
1 6.25 3.14  
2 3.75 1.12  
3 4.10 1.80  
4 4.36 5.21  
5 3.90 4.27
```

```
$clusters[[11]]  
      X1  X2
```



```

1 0.89 2.94
2 6.25 3.14
3 3.75 1.12
4 4.10 1.80
5 4.36 5.21
6 3.90 4.27

```

```

$groupedClusters
  cluster1 cluster2
1         3         5
2         2         6
3         7         8
4         4         9
5         1        10

```

■ : Explicacion

```
> agglomerativeHC.details(m, 'EUC', 'MIN')
```

```

[[1]]
      [,1] [,2] [,3]
[1,] 0.89 2.94    1

```

```

[[2]]
      [,1] [,2] [,3]
[1,] 4.36 5.21    1

```

```

[[3]]
      [,1] [,2] [,3]
[1,] 3.75 1.12    1

```

```

[[4]]
      [,1] [,2] [,3]
[1,] 6.25 3.14    1

```

```

[[5]]
      [,1] [,2] [,3]
[1,] 4.1  1.8    1

```

```

[[6]]
      [,1] [,2] [,3]
[1,] 3.9  4.27    1

```

```

      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,] 0.000000 4.146541 3.3899853 5.363730 3.4064204 3.290745
[2,] 4.146541 0.000000 4.1352388 2.803034 3.4198977 1.046518
[3,] 3.389985 4.135239 0.0000000 3.214094 0.7647876 3.153569
[4,] 5.363730 2.803034 3.2140940 0.000000 2.5333969 2.607566

```

```

[5,] 3.406420 3.419898 0.7647876 2.533397 0.0000000 2.478084
[6,] 3.290745 1.046518 3.1535694 2.607566 2.4780839 0.0000000
      X1      X2
1 3.75 1.12
2 4.10 1.80
      [,1]      [,2] [,3]      [,4] [,5]      [,6]      [,7]
[1,] 0.000000 4.146541      0 5.363730      0 3.290745 3.389985
[2,] 4.146541 0.000000      0 2.803034      0 1.046518 3.419898
[3,] 0.000000 0.000000      0 0.000000      0 0.000000 0.000000
[4,] 5.363730 2.803034      0 0.000000      0 2.607566 2.533397
[5,] 0.000000 0.000000      0 0.000000      0 0.000000 0.000000
[6,] 3.290745 1.046518      0 2.607566      0 0.000000 2.478084
[7,] 3.389985 3.419898      0 2.533397      0 2.478084 0.000000
      X1      X2
1 4.36 5.21
2 3.90 4.27
      [,1] [,2] [,3]      [,4] [,5] [,6]      [,7]      [,8]
[1,] 0.000000      0      0 5.363730      0      0 3.389985 3.290745
[2,] 0.000000      0      0 0.000000      0      0 0.000000 0.000000
[3,] 0.000000      0      0 0.000000      0      0 0.000000 0.000000
[4,] 5.363730      0      0 0.000000      0      0 2.533397 2.607566
[5,] 0.000000      0      0 0.000000      0      0 0.000000 0.000000
[6,] 0.000000      0      0 0.000000      0      0 0.000000 0.000000
[7,] 3.389985      0      0 2.533397      0      0 0.000000 2.478084
[8,] 3.290745      0      0 2.607566      0      0 2.478084 0.000000
      X1      X2
1 3.75 1.12
2 4.10 1.80
3 4.36 5.21
4 3.90 4.27
      [,1] [,2] [,3]      [,4] [,5] [,6] [,7] [,8]      [,9]
[1,] 0.000000      0      0 5.363730      0      0      0      0 3.290745
[2,] 0.000000      0      0 0.000000      0      0      0      0 0.000000
[3,] 0.000000      0      0 0.000000      0      0      0      0 0.000000
[4,] 5.363730      0      0 0.000000      0      0      0      0 2.533397
[5,] 0.000000      0      0 0.000000      0      0      0      0 0.000000
[6,] 0.000000      0      0 0.000000      0      0      0      0 0.000000
[7,] 0.000000      0      0 0.000000      0      0      0      0 0.000000
[8,] 0.000000      0      0 0.000000      0      0      0      0 0.000000
[9,] 3.290745      0      0 2.533397      0      0      0      0 0.000000
      X1      X2
1 6.25 3.14
2 3.75 1.12
3 4.10 1.80
4 4.36 5.21
5 3.90 4.27
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]      [,10]
[1,] 0.000000      0      0      0      0      0      0      0      0 3.290745

```

```

[2,] 0.000000 0 0 0 0 0 0 0 0 0.000000
[3,] 0.000000 0 0 0 0 0 0 0 0 0.000000
[4,] 0.000000 0 0 0 0 0 0 0 0 0.000000
[5,] 0.000000 0 0 0 0 0 0 0 0 0.000000
[6,] 0.000000 0 0 0 0 0 0 0 0 0.000000
[7,] 0.000000 0 0 0 0 0 0 0 0 0.000000
[8,] 0.000000 0 0 0 0 0 0 0 0 0.000000
[9,] 0.000000 0 0 0 0 0 0 0 0 0.000000
[10,] 3.290745 0 0 0 0 0 0 0 0 0.000000

```

```

      X1  X2
1 0.89 2.94
2 6.25 3.14
3 3.75 1.12
4 4.10 1.80
5 4.36 5.21
6 3.90 4.27

```

■ : Explicacion

```
> agglomerativeHC.details(m, 'EUC', 'MAX')
```

```

[[1]]
      [,1] [,2] [,3]
[1,] 0.89 2.94 1

```

```

[[2]]
      [,1] [,2] [,3]
[1,] 4.36 5.21 1

```

```

[[3]]
      [,1] [,2] [,3]
[1,] 3.75 1.12 1

```

```

[[4]]
      [,1] [,2] [,3]
[1,] 6.25 3.14 1

```

```

[[5]]
      [,1] [,2] [,3]
[1,] 4.1 1.8 1

```

```

[[6]]
      [,1] [,2] [,3]
[1,] 3.9 4.27 1

```

```

      [,1]      [,2]      [,3]      [,4]      [,5]      [,6]
[1,] 0.000000 4.146541 3.3899853 5.363730 3.4064204 3.290745
[2,] 4.146541 0.000000 4.1352388 2.803034 3.4198977 1.046518
[3,] 3.389985 4.135239 0.0000000 3.214094 0.7647876 3.153569

```

```
[4,] 5.363730 2.803034 3.2140940 0.000000 2.5333969 2.607566
[5,] 3.406420 3.419898 0.7647876 2.533397 0.0000000 2.478084
[6,] 3.290745 1.046518 3.1535694 2.607566 2.4780839 0.000000
```

```
      X1    X2
1 3.75 1.12
2 4.10 1.80
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7]
[1,] 0.000000 4.146541 0 5.363730 0 3.290745 3.406420
[2,] 4.146541 0.000000 0 2.803034 0 1.046518 4.135239
[3,] 0.000000 0.000000 0 0.000000 0 0.000000 0.000000
[4,] 5.363730 2.803034 0 0.000000 0 2.607566 3.214094
[5,] 0.000000 0.000000 0 0.000000 0 0.000000 0.000000
[6,] 3.290745 1.046518 0 2.607566 0 0.000000 3.153569
[7,] 3.406420 4.135239 0 3.214094 0 3.153569 0.000000
```

```
      X1    X2
1 4.36 5.21
2 3.90 4.27
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8]
[1,] 0.000000 0 0 5.363730 0 0 3.406420 4.146541
[2,] 0.000000 0 0 0.000000 0 0 0.000000 0.000000
[3,] 0.000000 0 0 0.000000 0 0 0.000000 0.000000
[4,] 5.363730 0 0 0.000000 0 0 3.214094 2.803034
[5,] 0.000000 0 0 0.000000 0 0 0.000000 0.000000
[6,] 0.000000 0 0 0.000000 0 0 0.000000 0.000000
[7,] 3.406420 0 0 3.214094 0 0 0.000000 4.135239
[8,] 4.146541 0 0 2.803034 0 0 4.135239 0.000000
```

```
      X1    X2
1 6.25 3.14
2 4.36 5.21
3 3.90 4.27
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9]
[1,] 0.000000 0 0 0 0 0 3.406420 0 5.363730
[2,] 0.000000 0 0 0 0 0 0.000000 0 0.000000
[3,] 0.000000 0 0 0 0 0 0.000000 0 0.000000
[4,] 0.000000 0 0 0 0 0 0.000000 0 0.000000
[5,] 0.000000 0 0 0 0 0 0.000000 0 0.000000
[6,] 0.000000 0 0 0 0 0 0.000000 0 0.000000
[7,] 3.40642 0 0 0 0 0 0.000000 0 4.135239
[8,] 0.000000 0 0 0 0 0 0.000000 0 0.000000
[9,] 5.36373 0 0 0 0 0 4.135239 0 0.000000
```

```
      X1    X2
1 0.89 2.94
2 3.75 1.12
3 4.10 1.80
```

```
      [,1] [,2] [,3] [,4] [,5] [,6] [,7] [,8] [,9] [,10]
[1,] 0 0 0 0 0 0 0 0 0 0.000000 0.000000
[2,] 0 0 0 0 0 0 0 0 0 0.000000 0.000000
[3,] 0 0 0 0 0 0 0 0 0 0.000000 0.000000
```

```

[4,]    0    0    0    0    0    0    0    0    0 0.00000 0.00000
[5,]    0    0    0    0    0    0    0    0    0 0.00000 0.00000
[6,]    0    0    0    0    0    0    0    0    0 0.00000 0.00000
[7,]    0    0    0    0    0    0    0    0    0 0.00000 0.00000
[8,]    0    0    0    0    0    0    0    0    0 0.00000 0.00000
[9,]    0    0    0    0    0    0    0    0    0 0.00000 5.36373
[10,]   0    0    0    0    0    0    0    0    0 5.36373 0.00000

```

```

      X1  X2
1 6.25 3.14
2 4.36 5.21
3 3.90 4.27
4 0.89 2.94
5 3.75 1.12
6 4.10 1.80

```

■ : Explicacion

>

### 2.1.2. Clusterización Jerárquica Aglomerativa

El segundo conjunto de datos, que se empleará para realizar el análisis de clasificación no supervisada con Clusterización Jerárquica Aglomerativa, estará formado por 6 calificaciones de estudiantes: 1.{0.89, 2.94}; 2.{4.36, 5.21}; 3.{3.75, 1.12}; 4.{6.25, 3.14}; 5.{4.1, 1.8}; 6.{3.9, 4.27}.

### Solución

## 2.2. Análisis de clasificación supervisada

### 2.2.1. Árboles de decisión

El tercer conjunto de datos, que se empleará para realizar el análisis de clasificación supervisada utilizando árboles de decisión, estará formado por las siguientes 9 calificaciones de estudiantes: 1. {A,A,B,Ap}; 2. {A,B,D,Ss}; 3. {D,D,C,Ss}; 4. {D,D,A,Ss}; 5. {B,C,B,Ss}; 6. {C,B,B,Ap}; 7. {B,B,A,Ap}; 8. {C,D,C,Ss}; 9. {B,A,C,Ss}, donde las características de las calificaciones son: {Teoría, Laboratorio, Prácticas, Calificación Global).

### 2.2.2. Regresión

El cuarto conjunto de datos, que se empleará para realizar el análisis de clasificación supervisada utilizando regresión, estará formado por los siguientes 4 radios ecuatoriales y densidades de los planetas interiores: {Mercurio,2.4,5.4; Venus,6.1,5.2; Tierra,6.4,5.5; Marte,3.4,3.9}

### 3. Ejercicios de forma autónoma

Realización de cuatro ejercicios de forma autónoma por cada grupo de estudiantes en los que se van a realizar, utilizando el entorno R, dos análisis de clasificación no supervisada y dos análisis de clasificación supervisada, aplicando todos los conceptos teóricos vistos en cada lección.

#### 3.1. Análisis de clasificación no supervisada

##### 3.1.1. K-means

El primer conjunto de datos, que se empleará para realizar el análisis de clasificación no supervisada con K-means, estará formado por los siguientes 15 valores de velocidades de respuesta y temperaturas normalizadas de un microprocesador {Velocidad, Temperatura}: 1.{3.5, 4.5}; 2.{0.75, 3.25}; 3.{0, 3}; 4.{1.75, 0.75}; 5.{3, 3.75}; 6.{3.75, 4.5}; 7.{1.25, 0.75}; 8.{0.25, 3}; 9.{3.5, 4.25}; 10.{1.5, 0.5}; 11.{1, 1}; 12.{3, 4}; 13.{0.5, 3}; 14.{2, 0.25}; 15.{0, 2.5}. Del análisis visual de los datos se ha concluido que hay una alta probabilidad que sean tres clusters.

**Solución:**

##### 3.1.2. Clusterización Jerárquica Aglomerativa

El segundo conjunto de datos, que se empleará para realizar el análisis de clasificación no supervisada con Clusterización Jerárquica Aglomerativa, será el mismo que el utilizado en el ejercicio anterior, por lo tanto estará formado por los siguientes 15 valores de velocidades de respuesta y temperaturas normalizadas de un microprocesador {Velocidad, Temperatura}: 1.{3.5, 4.5}; 2.{0.75, 3.25}; 3.{0, 3}; 4.{1.75, 0.75}; 5.{3, 3.75}; 6.{3.75, 4.5}; 7.{1.25, 0.75}; 8.{0.25, 3}; 9.{3.5, 4.25}; 10.{1.5, 0.5}; 11.{1, 1}; 12.{3, 4}; 13.{0.5, 3}; 14.{2, 0.25}; 15.{0, 2.5}. Del análisis visual de los datos se ha concluido que hay una alta probabilidad que sean tres clusters.

**Solución:**

- `datos <- read.csv("Datos2.2.csv")`: Se leen los datos del archivo .csv y se guardan en la variable "datos".
  - `(datos<-t(datos))`: Se hace la traspuesta de la matriz datos para un mejor tratamiento.
  - `(clMin<-agglomerative_clustering(datos,"single",FALSE,FALSE))` : *Clusterización usando*
- Parámetros:
- `datos`: Variable que contiene los puntos a evaluar.
  - `"single"`: Algoritmo de clusterización, en este caso usando minimum linkage.
  - `details="FALSE"`: Booleano para determinar si se imprimen o no los logs con la explicación de todo el proceso.

- `waiting="FALSE"`: Booleano para determinar si tiene que esperar input del usuario para ir imprimiendo los logs.

- Salida: Matriz con los clusters y sus distancias.
- Explicación: Realiza un análisis jerárquico aglomerativo de los datos introducidos por parámetro. Repite una serie de pasos hasta que solo quede un cluster. Inicialmente cada punto se asigna a su propio cluster y se calcula la matriz de distancias entre ellos. Después se calcula la proximidad entre dos clusters con la distancia entre los 2 puntos más cercanos de dichos clusters  $\min\{d(x, y) : x \in A, y \in B\}$

```
> library("clustlearn")
> datos <- read.csv("Datos2.2.csv")
> (datos<-t(datos))
\end{Sinput}
\begin{Soutput}
      [,1]
X3.5    4.50
X0.75   3.25
X0       3.00
X1.75   0.75
X3       3.75
X3.75   4.50
X1.25   0.75
X0.25   3.00
X3.5.1  4.25
X1.5    0.50
X1       1.00
X3.1    4.00
X0.5    3.00
X2       0.25
X0.1    2.50
\end{Soutput}
\begin{Sinput}
> (clMin<-agglomerative_clustering(datos,"single",FALSE,FALSE))
\end{Sinput}
\begin{Soutput}
Cluster method   : single
Distance         : Euclidean
Number of objects: 15
\end{Soutput}
```

`(clMax<-agglomerative_clustering(datos,"complete",FALSE,FALSE))` : *Clusterizacin usando complete linkage*  
 Parámetros:

- `datos`: Variable que contiene los puntos a evaluar.
- `"single"`: Algoritmo de clusterización, en este caso usando maximum linkage.



- details="FALSE": Booleano para determinar si se imprimen o no los logs con la explicación de todo el proceso.
- waiting="FALSE": Booleano para determinar si tiene que esperar input del usuario para ir imprimiendo los logs.

Salida: Matriz con los clusters y sus distancias.

Explicación: Realiza un análisis jerárquico aglomerativo de los datos introducidos por parámetro. Repite una serie de pasos hasta que solo quede un cluster. Inicialmente cada punto se asigna a su propio cluster y se calcula la matriz de distancias entre ellos. Después se calcula la proximidad entre dos clusters con la distancia entre los puntos más lejanos de dichos clusters  $\max\{d(x,y) : x \in A, y \in B\}$

```
> library("clustlearn")
> datos <- read.csv("Datos2.2.csv")
> (datos<-t(datos))
\end{Sinput}
\begin{Soutput}
      [,1]
X3.5    4.50
X0.75   3.25
X0       3.00
X1.75   0.75
X3       3.75
X3.75   4.50
X1.25   0.75
X0.25   3.00
X3.5.1  4.25
X1.5    0.50
X1       1.00
X3.1    4.00
X0.5    3.00
X2       0.25
X0.1    2.50
\end{Soutput}
\begin{Sinput}
> (clMin<-agglomerative_clustering(datos,"complete",FALSE,FALSE))
\end{Sinput}
\begin{Soutput}
Cluster method      : complete
Distance            : Euclidean
Number of objects: 15
\end{Soutput}
\end{Schunk}

\end{itemize}

\item \texttt{(clAvg<-agglomerative_clustering(datos,"average",FALSE,1
```

```

\begin{itemize}
  \item [-] Parámetros:
  \begin{itemize}
    \item \texttt{datos}: Variable que contiene los puntos
    \item \texttt{"average"}: Algoritmo de clusterización
    \item \texttt{details="FALSE"}: Booleano para determinar
    \item \texttt{waiting="FALSE"}: Booleano para determinar
  \end{itemize}
  \item [-] Salida: Matriz con los clusters y sus distancias.
  \item [-] Explicación: Realiza un análisis jerárquico aglomerativo
\begin{Schunk}
\begin{Sininput}
> library("clustlearn")
> datos <- read.csv("Datos2.2.csv")
> (datos<-t(datos))
\end{Sininput}
\begin{Soutput}
      [,1]
X3.5    4.50
X0.75   3.25
X0       3.00
X1.75   0.75
X3       3.75
X3.75   4.50
X1.25   0.75
X0.25   3.00
X3.5.1  4.25
X1.5    0.50
X1       1.00
X3.1    4.00
X0.5    3.00
X2       0.25
X0.1    2.50
\end{Soutput}
\begin{Sininput}
> (clMin<-agglomerative_clustering(datos,"average",FALSE,FALSE))
\end{Sininput}
\begin{Soutput}
Cluster method   : average
Distance         : Euclidean
Number of objects: 15
\end{Soutput}
\end{Schunk}
\end{itemize}
\end{itemize}

\subsection{Análisis de clasificación supervisada}

```

```
\subsubsection{Árboles de decisión}
```

El tercer conjunto de datos, que se empleará para realizar el análisis de clas

```
\subsubsection{Regresión}
```

El cuarto conjunto de datos, que se empleará para realizar el análisis de clas

```
\paragraph{Solución:}
```

Algoritmo de minería de reglas de asociación (apriori): Su ob

```
\begin{itemize}
```

```
\item[-] Parámetros:
```

```
\begin{itemize}
```

```
\item \texttt{transacciones}: Lista de sucesos
```

```
\item \texttt{soporte}: Umbral mínimo de sopor
```

```
\item \texttt{confianza}: Umbral mínimo de con
```

```
\end{itemize}
```

```
\item[-] Retorno:
```

```
\item[-] Explicación:
```

```
\end{itemize}
```

```
\begin{Schunk}
```

```
\begin{Sinput}
```

```
>
```

```
\end{Sinput}
```

```
\end{Schunk}
```