Practica 4: Fundamentos de la Ciencia de Datos

Daniel Lopez Moreno Alejandro Fernandez Maceira
Alvaro Maestre Santa

November 22, 2019

1 Clasificación no supervisada

En este apartado vamos a realizar una clasificación no supervisada con unos datos proporcionados por el profesor. Para ello usaremos el algoritmo K-means, sobre la muestra utilizada en clase, a parte se deberá obtener **los centroides de los clusters obtenidos**.
En primer lugar, debemos tener la librería **stats**, si no la tenemos, tenemos que instalarla.

Ahora procedemos a cargar los datos en una matriz para posteriormente hacer la transpuesta y hacer el algoritmo **K-means**, como ya he mencionado antes, los datos que cargamos en la matriz, están proporcionados por el profesor:

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

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

Ahora calculamos la transpuesta de la matriz:

```
> (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
```

A continuación, cargamos otra matriz de datos, y posteriormente haremos la transpuesta:

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

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

Como ya hemos mencionado, ahora procedemos a calcular la transpuesta de la matriz cargada posteriormente:

```
> (c<-t(c))
```

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

Ahora procedemos a realizar el algoritmo **K-means** para la clasificación, en este comando, tenemos que meter las dos transpuestas calculadas previamente y posteriormente ponemos el número de iteraciones que queremos que haga **K-means**, en este caso 4, cuando llegue a la 4 iteración parará. El resultado obtenido es el siguiente:

```
> (clasificaciones<-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"
```

Ahora vamos a proceder a explicar los resultados obtenidos:

- **En primer lugar**: tenemos dos clusters de 4 y 4 datos cada uno para un total de 8 registros.
- **cluster means**: en este apartado, el algoritmo calcula la medida más óptima para hallar los centroides, los cuales, muestra en **Cluster Means**, cuando detecta estos, en este caso, detecta 2 centroides.
- **Clustering vector**: indica el pronóstico para cada registro testeado con el algoritmo.
- **[1] 3.75 2.75**: estos valores, es la inercia intra-clúster de cada grupo.

- **between_SS / total_SS**: es una medida de calidad e indica que tanto están separados los grupos de manera inter-cluster en relación al agrupamiento intra-cluster, mientras, se esté más cercano al 100
- **Available components**: tenemos los elementos disponibles del modelo, los cuales, vamos a explicar a continuación:
  - **Cluster**: La categorización asignada a cada observación de los datos introducidos o dataset en función a su cercanía a estos centros.
  - **Centers**: Los centroides.
  - **Totss**: Inercia total del conjunto de datos.
  - **Withinss**: Inercia intra-clases de cada uno de los grupos.
  - **Tot.withinss**: Inercia intra-clases total.
  - **Betweenss**: Inercia inter-clases.
  - **Size**: El tamaño de cada grupo.
  - **Iter**: Número de iteraciones empleado.

Ahora unimos, con la función **cbind**, a la matriz de datos(clasifiaciones), en este caso, une el vector con la matriz m:

```
> (m = cbind(clasificaciones$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
```

Ahora con la funcion **subset**, cogemos de la columna 1, aquellos valores que sean igual a 1, y creamos una matriz con las filas cuyo valor de la columna 1 es 1:

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

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

Hacemos lo mismo con la columna 1, pero que contengan el valor 2:

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

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

Quitamos la primera columna de las dos matrices generadas anteriormente, quedando los siguientes resultados:

```
> (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
```

Estas dos matrices, son las matrices de los centroides resultantes.

2 Desarrollo por parte del grupo

En este apartado vamos a realizar una **Clasificación no supervisada** sobre una base de datos que reune información de los personajes de la saga de películas **Star Wars**. Este archivo **"characters.csv"** está compuesto 87 personajes con atributos repartidos en 10 columnas:

- **Name**: (Nombre)
- **Height**: (Altura)
- **Mass**: (Peso)
- **Hair_color**: (Color de Pelo)
- **Skin_color**: (Color de Piel)
- **Eye_color**: (Color de Ojos)
- **Birth_year**: (Año de nacimiento)
- **Gender**: (Genero)
- **Homeworld**: (Hogar)
- **Species**: (Especie)

Nuestro análisis de clasificación se centrará en *clusterizar* los personajes según su **altura y peso**. Para empezar leeremos nuestro archivo .csv y lo almacenaremos en nuestra variable **personajes**:

```
> personajes<- read.csv("characters.csv")
```

Algunos de nuestros personajes no tienen la suficiente información sobre su altura o peso como para clasificarlos, y en estos atributos cuentan con el valor **NA**. Como no tenemos información sobre estos personajes, no podemos clasificarlos, asi que debemos eliminarlos de nuestra matriz utilizando la funcion **complete.cases** y especificando que queremos trabajar sobre las columnas de altura y peso (columnas 2 y 3), el resto no nos importa que tengan NA:

```
> (personajes <- personajes[complete.cases(personajes[, 2:3]),])
```

```
                   name height  mass    hair_color          skin_color
1         Luke Skywalker    172    77         blond                fair
2                  C-3PO    167    75          <NA>                gold
3                  R2-D2     96    32          <NA>         white, blue
4            Darth Vader    202   136          none               white
5            Leia Organa    150    49         brown               light
6              Owen Lars    178   120   brown, grey               light
7     Beru Whitesun lars    165    75         brown               light
8                  R5-D4     97    32          <NA>          white, red
9      Biggs Darklighter    183    84         black               light
10        Obi-Wan Kenobi    182    77 auburn, white                fair
11      Anakin Skywalker    188    84         blond                fair
13             Chewbacca    228   112         brown                <NA>
14              Han Solo    180    80         brown                fair
15                Greedo    173    74          <NA>               green
16 Jabba Desilijic Tiure    175 1,358          <NA>    green-tan, brown
17        Wedge Antilles    170    77         brown                fair
18      Jek Tono Porkins    180   110         brown                fair
19                  Yoda     66    17         white               green
20             Palpatine    170    75          grey                pale
21             Boba Fett    183  78.2         black                fair
22                 IG-88    200   140          none               metal
23                 Bossk    190   113          none               green
24      Lando Calrissian    177    79         black                dark
25                 Lobot    175    79          none               light
26                Ackbar    180    83          none        brown mottle
29 Wicket Systri Warrick     88    20         brown               brown
30             Nien Nunb    160    68          none                grey
31          Qui-Gon Jinn    193    89         brown                fair
32           Nute Gunray    191    90          none       mottled green
34         Jar Jar Binks    196    66          none              orange
35          Roos Tarpals    224    82          none                grey
39               Sebulba    112    40          none           grey, red
42            Darth Maul    175    80          none                 red
44           Ayla Secura    178    55          none                blue
45              Dud Bolt     94    45          none          blue, grey
47        Ben Quadinaros    163    65          none grey, green, yellow
48            Mace Windu    188    84          none                dark
49          Ki-Adi-Mundi    198    82         white                pale
50             Kit Fisto    196    87          none               green
52            Adi Gallia    184    50          none                dark
```

55	Plo Koon	188	80	none	orange	
57	Gregar Typho	185	85	black	dark	
60	Poggle the Lesser	183	80	none	green	
61	Luminara Unduli	170	56.2	black	yellow	
62	Barriss Offee	166	50	black	yellow	
64	Dooku	193	80	white	fair	
66	Jango Fett	183	79	black	tan	
67	Zam Wesell	168	55	blonde	fair, green, yellow	
68	Dexter Jettster	198	102	none	brown	
69	Lama Su	229	88	none	grey	
72	Ratts Tyerell	79	15	none	grey, blue	
74	Wat Tambor	193	48	none	green, grey	
76	Shaak Ti	178	57	none	red, blue, white	
77	Grievous	216	159	none	brown, white	
78	Tarfful	234	136	brown	brown	
79	Raymus Antilles	188	79	brown	light	
80	Sly Moore	178	48	none	pale	
81	Tion Medon	206	80	none	grey	
87	PadmÃ© Amidala	165	45	brown	light	

	eye_color	birth_year	gender	homeworld	species
1	blue	19BBY	male	Tatooine	Human
2	yellow	112BBY	<NA>	Tatooine	Droid
3	red	33BBY	<NA>	Naboo	Droid
4	yellow	41.9BBY	male	Tatooine	Human
5	brown	19BBY	female	Alderaan	Human
6	blue	52BBY	male	Tatooine	Human
7	blue	47BBY	female	Tatooine	Human
8	red	<NA>	<NA>	Tatooine	Droid
9	brown	24BBY	male	Tatooine	Human
10	blue-gray	57BBY	male	Stewjon	Human
11	blue	41.9BBY	male	Tatooine	Human
13	blue	200BBY	male	Kashyyyk	Wookiee
14	brown	29BBY	male	Corellia	Human
15	black	44BBY	male	Rodia	Rodian
16	orange	600BBY	hermaphrodite	Nal Hutta	Hutt
17	hazel	21BBY	male	Corellia	Human
18	blue	<NA>	male	Bestine IV	Human
19	brown	896BBY	male	<NA>	Yoda's species
20	yellow	82BBY	male	Naboo	Human
21	brown	31.5BBY	male	Kamino	Human
22	red	15BBY	none	<NA>	Droid
23	red	53BBY	male	Trandosha	Trandoshan
24	brown	31BBY	male	Socorro	Human
25	blue	37BBY	male	Bespin	Human
26	orange	41BBY	male	Mon Cala	Mon Calamari
29	brown	8BBY	male	Endor	Ewok
30	black	<NA>	male	Sullust	Sullustan
31	blue	92BBY	male	<NA>	Human
32	red	<NA>	male	Cato Neimoidia	Neimodian
34	orange	52BBY	male	Naboo	Gungan

```
35        orange       <NA>         male          Naboo         Gungan
39        orange       <NA>         male      Malastare            Dug
42        yellow      54BBY         male       Dathomir         Zabrak
44         hazel      48BBY       female         Ryloth       Twi'lek
45        yellow       <NA>         male        Vulpter     Vulptereen
47        orange       <NA>         male           Tund          Toong
48         brown      72BBY         male     Haruun Kal          Human
49        yellow      92BBY         male          Cerea        Cerean
50         black       <NA>         male    Glee Anselm       Nautolan
52          blue       <NA>       female      Coruscant     Tholothian
55         black      22BBY         male          Dorin        Kel Dor
57         brown       <NA>         male          Naboo          Human
60        yellow       <NA>         male       Geonosis      Geonosian
61          blue      58BBY       female         Mirial       Mirialan
62          blue      40BBY       female         Mirial       Mirialan
64         brown     102BBY         male        Serenno          Human
66         brown      66BBY         male   Concord Dawn          Human
67        yellow       <NA>       female          Zolan       Clawdite
68        yellow       <NA>         male           Ojom       Besalisk
69         black       <NA>         male         Kamino       Kaminoan
72          <NA>       <NA>         male    Aleen Minor         Aleena
74          <NA>       <NA>         male          Skako        Skakoan
76         black       <NA>       female          Shili        Togruta
77 green, yellow       <NA>         male          Kalee        Kaleesh
78          blue       <NA>         male       Kashyyyk        Wookiee
79         brown       <NA>         male       Alderaan          Human
80         white       <NA>       female         Umbara           <NA>
81         black       <NA>         male         Utapau         Pau'an
87         brown      46BBY       female          Naboo          Human
```

Ahora que ya tenemos nuestra matriz con los datos es hora de trabajar con ella. Crearemos una matriz a parte para aislar las medidas de altura y peso:

```
> height<-personajes$height
> mass<-personajes$mass
> (medidas<-cbind(height,mass))
```

```
      height mass
 [1,]    172   27
 [2,]    167   26
 [3,]     96   13
 [4,]    202    7
 [5,]    150   17
 [6,]    178    6
 [7,]    165   26
 [8,]     97   13
 [9,]    183   33
[10,]    182   27
[11,]    188   33
[12,]    228    4
[13,]    180   30
```

```
[14,]  173  25
[15,]  175   1
[16,]  170  27
[17,]  180   3
[18,]   66  11
[19,]  170  26
[20,]  183  28
[21,]  200   8
[22,]  190   5
[23,]  177  29
[24,]  175  29
[25,]  180  32
[26,]   88  12
[27,]  160  24
[28,]  193  37
[29,]  191  38
[30,]  196  23
[31,]  224  31
[32,]  112  14
[33,]  175  30
[34,]  178  19
[35,]   94  15
[36,]  163  22
[37,]  188  33
[38,]  198  31
[39,]  196  35
[40,]  184  18
[41,]  188  30
[42,]  185  34
[43,]  183  30
[44,]  170  20
[45,]  166  18
[46,]  193  30
[47,]  183  29
[48,]  168  19
[49,]  198   2
[50,]  229  36
[51,]   79   9
[52,]  193  16
[53,]  178  21
[54,]  216  10
[55,]  234   7
[56,]  188  29
[57,]  178  16
[58,]  206  30
[59,]  165  15
```

2.1 Clasificacion con k-means

Crearemos los centroides de nuestros clusters. Para diferenciarnos del ejercicio realizado en clase realizaremos la clasificación k-means separando en 3 clusters en vez de en 2. Los clusters empezarán siendo **C1(200cm, 100kg)**, **C2(150cm, 75kg)** y **C3(100cm, 50kg)**:

```
> centroides <-matrix(c(200,100,150,75,100,50),2,3)
> (centroides <- t(centroides))
```

```
     [,1] [,2]
[1,]  200  100
[2,]  150   75
[3,]  100   50
```

Una vez creados los centroides y aisladas nuestras medidas vamos a realizar la clasificación del mismo modo que la hemos realizado en clase, y posteriormente **utilizaremos otros métodos de clasificación para comparar** los resultados:

```
> (clasiKmeans<-kmeans(medidas,centroides,6))
```

```
K-means clustering with 3 clusters of sizes 9, 43, 7

Cluster means:
     height     mass
1 215.22222 15.00000
2 179.02326 24.34884
3  90.28571 12.42857

Clustering vector:
 [1] 2 2 3 1 2 2 2 3 2 2 2 1 2 2 2 2 2 3 2 2 1 2 2 2 2 3 2 2 2 2 1 3 2 2 3 2 2 2
[39] 2 2 2 2 2 2 2 2 2 2 2 1 1 3 2 2 1 1 2 2 1 2

Within cluster sum of squares by cluster:
[1] 2985.556 8258.744 1309.143
 (between_SS / total_SS =  84.2 %)

Available components:

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

Hemos observado que los centroides se han desplazado hacia los valores **C1(215.22222, 15.00000)**, **C2(179.02326, 24.34884)** y **C3(90.28571 12.42857)**. Después de clasificar mediante k-means, añadimos la columna de los clusters a la matriz de personajes y los separamos segun su cluster:

```
> personajesKmeans = cbind(clasiKmeans$cluster,personajes)
```

Cluster 1:

```
> (persc1 = subset(personajesKmeans,personajesKmeans[,1] == 1))
```

```
   clasiKmeans$cluster            name height mass hair_color   skin_color
4                    1     Darth Vader    202  136       none        white
13                   1       Chewbacca    228  112      brown         <NA>
22                   1           IG-88    200  140       none        metal
35                   1    Roos Tarpals    224   82       none         grey
68                   1 Dexter Jettster    198  102       none        brown
69                   1         Lama Su    229   88       none         grey
77                   1        Grievous    216  159       none brown, white
78                   1         Tarfful    234  136      brown        brown
81                   1      Tion Medon    206   80       none         grey
      eye_color birth_year gender homeworld  species
4        yellow    41.9BBY   male  Tatooine    Human
13         blue     200BBY   male  Kashyyyk  Wookiee
22          red      15BBY   none      <NA>    Droid
35       orange       <NA>   male     Naboo   Gungan
68       yellow       <NA>   male      Ojom Besalisk
69        black       <NA>   male    Kamino Kaminoan
77 green, yellow       <NA>   male     Kalee  Kaleesh
78         blue       <NA>   male  Kashyyyk  Wookiee
81        black       <NA>   male    Utapau   Pau'an
```

Cluster 2:

```
> (persc2 = subset(personajesKmeans,personajesKmeans[,1] == 2))
```

```
   clasiKmeans$cluster                 name height  mass    hair_color
1                    2        Luke Skywalker    172    77         blond
2                    2                C-3PO    167    75          <NA>
5                    2           Leia Organa    150    49         brown
6                    2             Owen Lars    178   120   brown, grey
7                    2    Beru Whitesun lars    165    75         brown
9                    2     Biggs Darklighter    183    84         black
10                   2        Obi-Wan Kenobi    182    77 auburn, white
11                   2      Anakin Skywalker    188    84         blond
14                   2              Han Solo    180    80         brown
15                   2                Greedo    173    74          <NA>
16                   2 Jabba Desilijic Tiure    175 1,358          <NA>
17                   2        Wedge Antilles    170    77         brown
18                   2      Jek Tono Porkins    180   110         brown
20                   2             Palpatine    170    75          grey
21                   2             Boba Fett    183  78.2         black
23                   2                Bossk    190   113          none
24                   2      Lando Calrissian    177    79         black
25                   2                Lobot    175    79          none
26                   2              Ackbar    180    83          none
30                   2            Nien Nunb    160    68          none
31                   2          Qui-Gon Jinn    193    89         brown
32                   2           Nute Gunray    191    90          none
34                   2         Jar Jar Binks    196    66          none
42                   2            Darth Maul    175    80          none
44                   2           Ayla Secura    178    55          none
```

47	2	Ben Quadinaros	163	65	none
48	2	Mace Windu	188	84	none
49	2	Ki-Adi-Mundi	198	82	white
50	2	Kit Fisto	196	87	none
52	2	Adi Gallia	184	50	none
55	2	Plo Koon	188	80	none
57	2	Gregar Typho	185	85	black
60	2	Poggle the Lesser	183	80	none
61	2	Luminara Unduli	170	56.2	black
62	2	Barriss Offee	166	50	black
64	2	Dooku	193	80	white
66	2	Jango Fett	183	79	black
67	2	Zam Wesell	168	55	blonde
74	2	Wat Tambor	193	48	none
76	2	Shaak Ti	178	57	none
79	2	Raymus Antilles	188	79	brown
80	2	Sly Moore	178	48	none
87	2	PadmÃ© Amidala	165	45	brown

	skin_color	eye_color	birth_year	gender	homeworld
1	fair	blue	19BBY	male	Tatooine
2	gold	yellow	112BBY	<NA>	Tatooine
5	light	brown	19BBY	female	Alderaan
6	light	blue	52BBY	male	Tatooine
7	light	blue	47BBY	female	Tatooine
9	light	brown	24BBY	male	Tatooine
10	fair	blue-gray	57BBY	male	Stewjon
11	fair	blue	41.9BBY	male	Tatooine
14	fair	brown	29BBY	male	Corellia
15	green	black	44BBY	male	Rodia
16	green-tan, brown	orange	600BBY	hermaphrodite	Nal Hutta
17	fair	hazel	21BBY	male	Corellia
18	fair	blue	<NA>	male	Bestine IV
20	pale	yellow	82BBY	male	Naboo
21	fair	brown	31.5BBY	male	Kamino
23	green	red	53BBY	male	Trandosha
24	dark	brown	31BBY	male	Socorro
25	light	blue	37BBY	male	Bespin
26	brown mottle	orange	41BBY	male	Mon Cala
30	grey	black	<NA>	male	Sullust
31	fair	blue	92BBY	male	<NA>
32	mottled green	red	<NA>	male	Cato Neimoidia
34	orange	orange	52BBY	male	Naboo
42	red	yellow	54BBY	male	Dathomir
44	blue	hazel	48BBY	female	Ryloth
47	grey, green, yellow	orange	<NA>	male	Tund
48	dark	brown	72BBY	male	Haruun Kal
49	pale	yellow	92BBY	male	Cerea
50	green	black	<NA>	male	Glee Anselm
52	dark	blue	<NA>	female	Coruscant
55	orange	black	22BBY	male	Dorin

```
57                dark     brown       <NA>         male          Naboo
60               green    yellow       <NA>         male       Geonosis
61              yellow      blue      58BBY       female         Mirial
62              yellow      blue      40BBY       female         Mirial
64                fair     brown     102BBY         male        Serenno
66                 tan     brown      66BBY         male   Concord Dawn
67 fair, green, yellow    yellow       <NA>       female          Zolan
74         green, grey      <NA>       <NA>         male          Skako
76    red, blue, white     black       <NA>       female          Shili
79                light     brown       <NA>         male       Alderaan
80                 pale     white       <NA>       female         Umbara
87                light     brown      46BBY       female          Naboo
        species
1         Human
2         Droid
5         Human
6         Human
7         Human
9         Human
10        Human
11        Human
14        Human
15       Rodian
16         Hutt
17        Human
18        Human
20        Human
21        Human
23   Trandoshan
24        Human
25        Human
26 Mon Calamari
30    Sullustan
31        Human
32    Neimodian
34       Gungan
42       Zabrak
44      Twi'lek
47        Toong
48        Human
49       Cerean
50     Nautolan
52   Tholothian
55      Kel Dor
57        Human
60    Geonosian
61     Mirialan
62     Mirialan
64        Human
66        Human
```

```
67     Clawdite
74      Skakoan
76      Togruta
79        Human
80         <NA>
87        Human
```

Cluster 3:

```
> (persc3 = subset(personajesKmeans,personajesKmeans[,1] == 3))
```

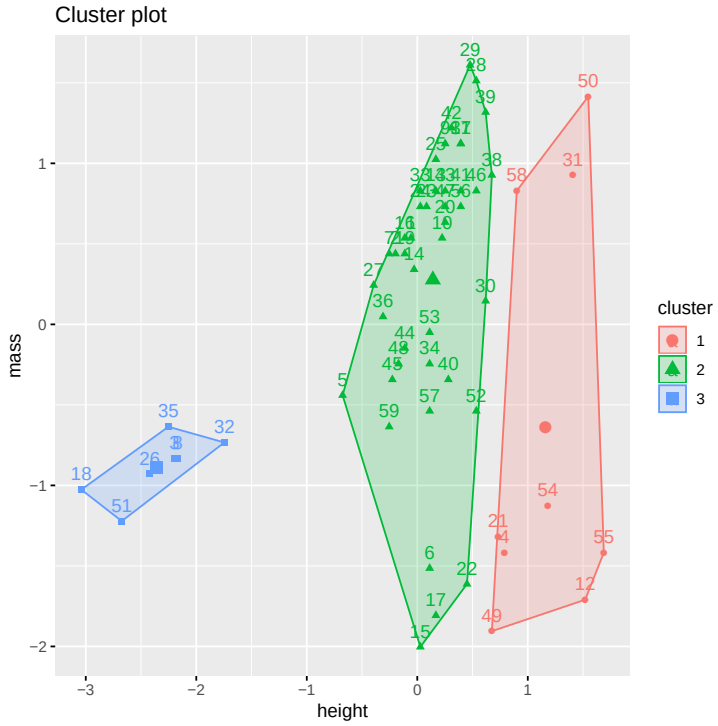
```
   clasiKmeans$cluster                  name height mass hair_color  skin_color
3                    3                R2-D2     96   32       <NA> white, blue
8                    3                R5-D4     97   32       <NA>  white, red
19                   3                 Yoda     66   17      white       green
29                   3 Wicket Systri Warrick     88   20      brown       brown
39                   3              Sebulba    112   40       none   grey, red
45                   3            Dud Bolt     94   45       none  blue, grey
72                   3         Ratts Tyerell     79   15       none  grey, blue
   eye_color birth_year gender   homeworld        species
3        red      33BBY   <NA>       Naboo          Droid
8        red       <NA>   <NA>    Tatooine          Droid
19     brown     896BBY   male        <NA> Yoda's species
29     brown       8BBY   male       Endor           Ewok
39    orange       <NA>   male   Malastare            Dug
45    yellow       <NA>   male     Vulpter     Vulptereen
72      <NA>       <NA>   male Aleen Minor         Aleena
```

Para poder ver con más detalle la división en clusters de los datos, se representarán gráficamente los clusters en dos gráficas distintas. Para ello harán falta las siguientes librerías.

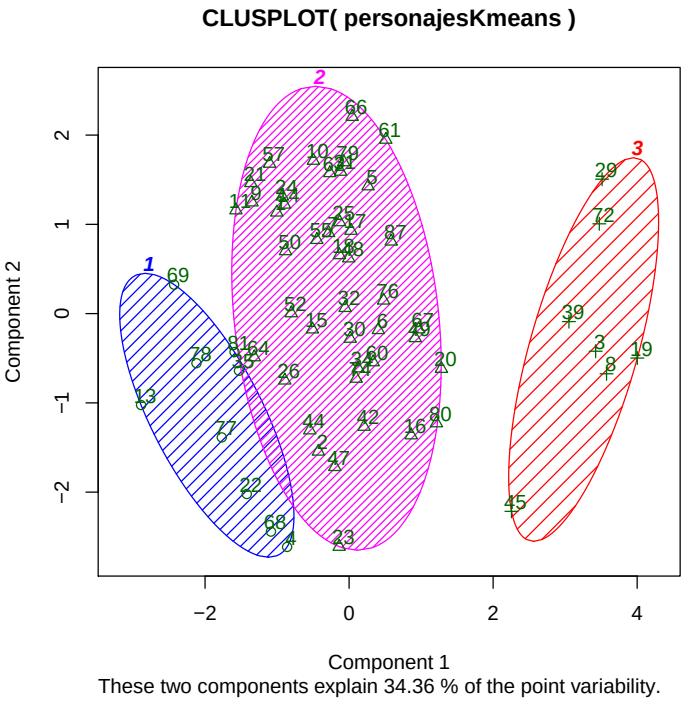
```
> install.packages("cluster")
> install.packages("factoextra")
> library(cluster)
> library(factoextra)
```

Ahora que están instalados los paquetes, se pueden representar los resultados anteriores.

```
> fviz_cluster(clasiKmeans,data=medidas)
```



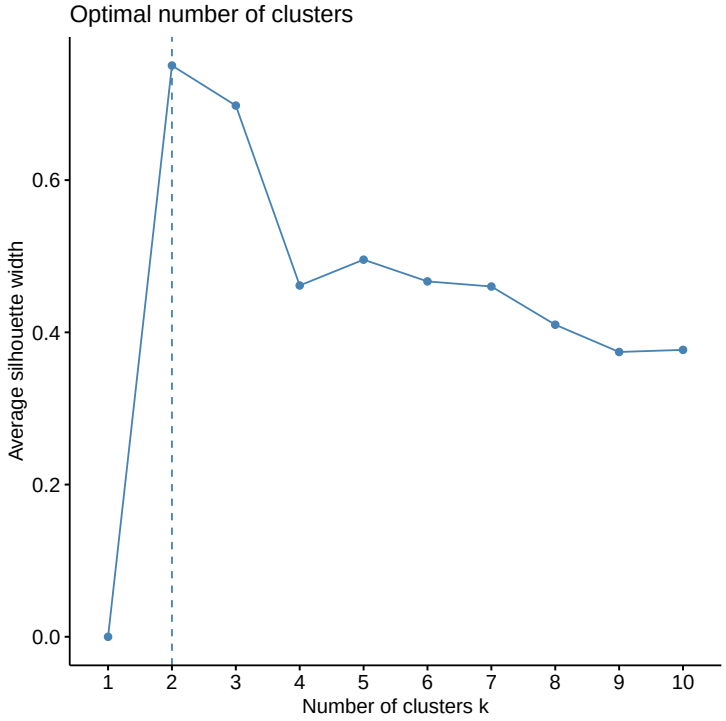

```
> clusplot(personajesKmeans,clasiKmeans$cluster,color=TRUE,shade=TRUE,labels=2,lines=0)
```



Una comprobación que se puede realizar para los datos es obtener el número

óptimo de clusters para clasificarlos. Para comprobarlo, se utilizan la medida de la silueta, que indica de forma aproximada cómo de bien se han clasificado los datos, es decir, si están en los clusters correctos.

```
> fviz_nbclust(medidas,kmeans,method="silhouette")
```



Como se puede observar, los mejores números de clusters para clasificar los datos están entre 2 y 3, con 1 cluster la clasificación es muy ineficiente y más clusters de 3 son innecesarios.

Como conclusión de esta clasificación de k-means podemos sacar que todos aquellos que quedan en el Cluster 1 comparten que su género es **masculino**. Los personajes pertenecientes al Cluster 3 también comparten que son de **género masculino**, a excepción de R2-D2 y R5-D4 que son droides de pequeña estatura. Sobre el Cluster 3 también cabe destacar que ninguno de los personajes es de Especie **humano**. Por último, en el Cluster 2 es donde se acumulan la mayor cantidad de personajes y donde es más dificil es diferenciarlos por atributos, aunque si se puede observar que **todos los humanos** (a excepción de Darth Vader) se encuentran en este cluster, rondando los 179cm y 24kg.

2.2 Clasificación mediante Clustering Jerarquico Aglomerativo

A continuación realizaremos la clasificación mediante **Clustering Jerarquico Aglomerativo** en inglés HAC. Mediante este método todos los personajes irán agrupandose en clusters de pequeño tamaño y poco a poco irán generando clusters mayores hasta lograr un solo cluster que agrupamiento todos los personajes. Como hemos aprendido en clase, el primer paso de este algoritmo es calcular la

matriz de distancias entre todos los personajes. Para ello utilizaremos la función **dist** y el parámetro **"euclidean"** para calcular las distancias mediante la formula euclidiana:

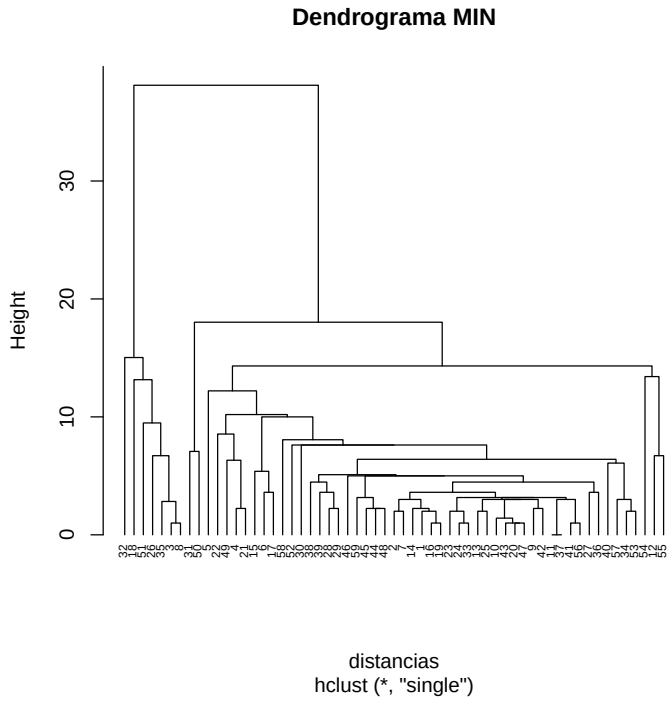
```
> distancias <- dist(medidas, method = "euclidean")
```

Una vez hemos calculado las distancias vamos a realizar la clasificación jerarquica. Existen tres métodos diferentes de agrupar en clusters: **MIN("single")**, **MAX("complete")** y **Group Average("average")**.

2.2.1 Enlace simple MIN

Realizaremos la clasificación utilizando la distancia **mínima** entre puntos a cada cluster:

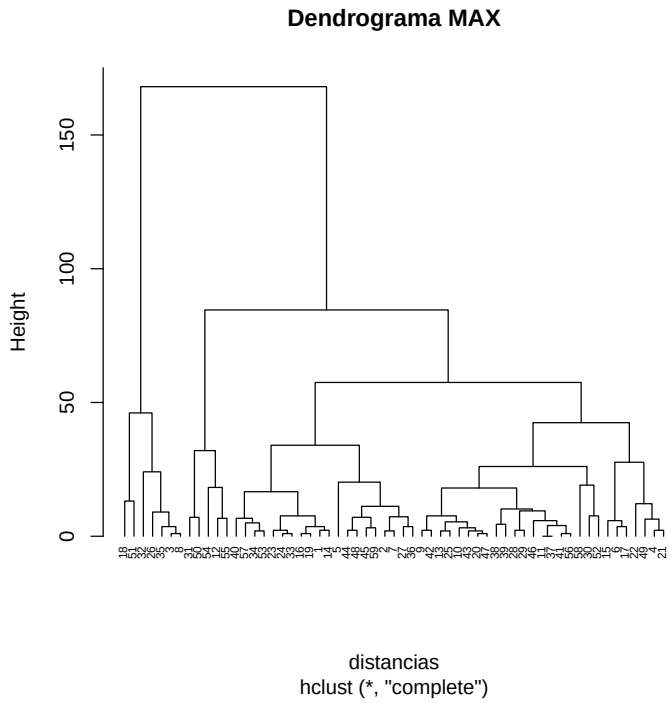
```
> hcmin <- hclust(distancias, method = "single" )
> plot(hcmin, cex = 0.6, hang = -1, main = "Dendrograma MIN")
```



2.2.2 Enlace completo MAX

Realizaremos la clasificación utilizando la distancia **máxima** entre puntos a cada cluster:

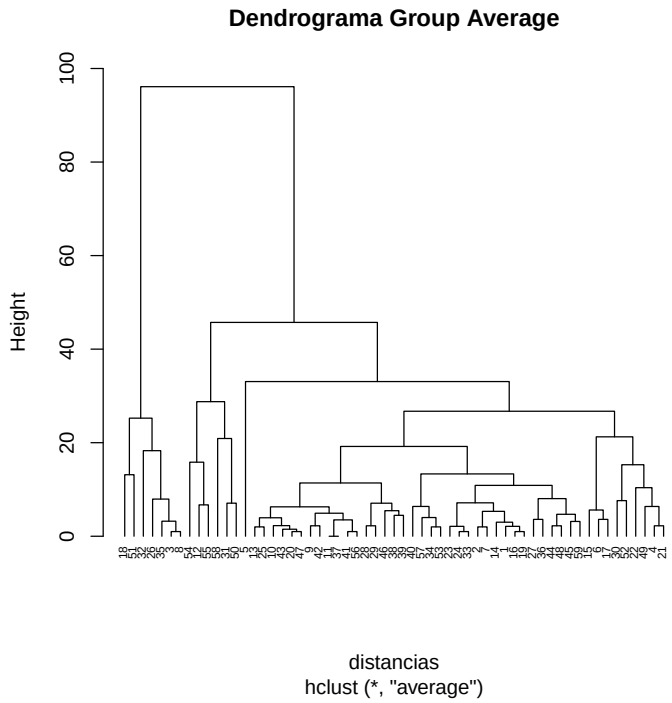
```
> hcmax <- hclust(distancias, method = "complete" )
> plot(hcmax, cex = 0.6, hang = -1, main = "Dendrograma MAX")
```



2.2.3 Enlace promedio Group Average

Realizaremos la clasificación utilizando la **media de las distancias** entre puntos a cada cluster:

```
> hcavg <- hclust(distancias, method = "average" )
> plot(hcavg, cex = 0.6, hang = -1, main = "Dendrograma Group Average")
```



Como se puede observar, todos los dendogramas son muy parecidos entre si, con ligeros cambios debido al método de agrupación escogido para cada uno.

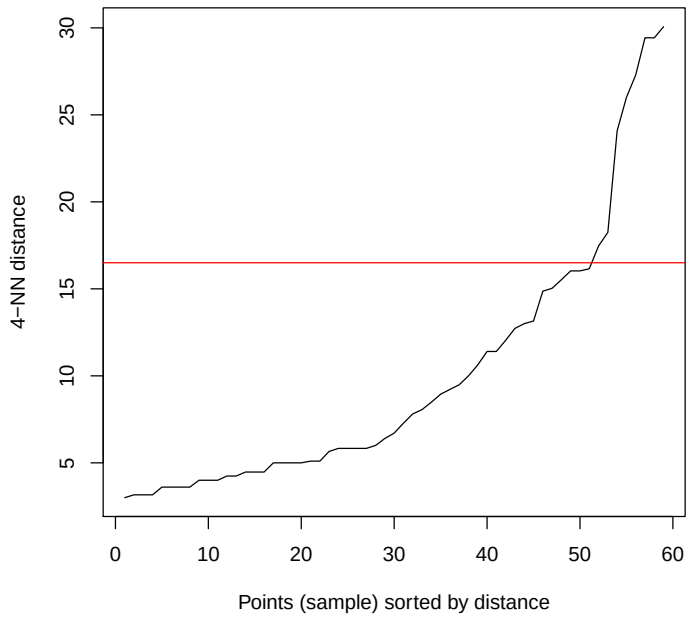
2.3 Clasificación mediante DBSCAN

Por último, realizaremos la clasificación mediante el algoritmo **DBSCAN**. Este algoritmo agrupará nuestros personajes en clusters basandose en la densidad, comenzando por una estimación de la distribución de densidad de cada personaje. Utilizaremos los paquetes **fpc** y **dbscan**:

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

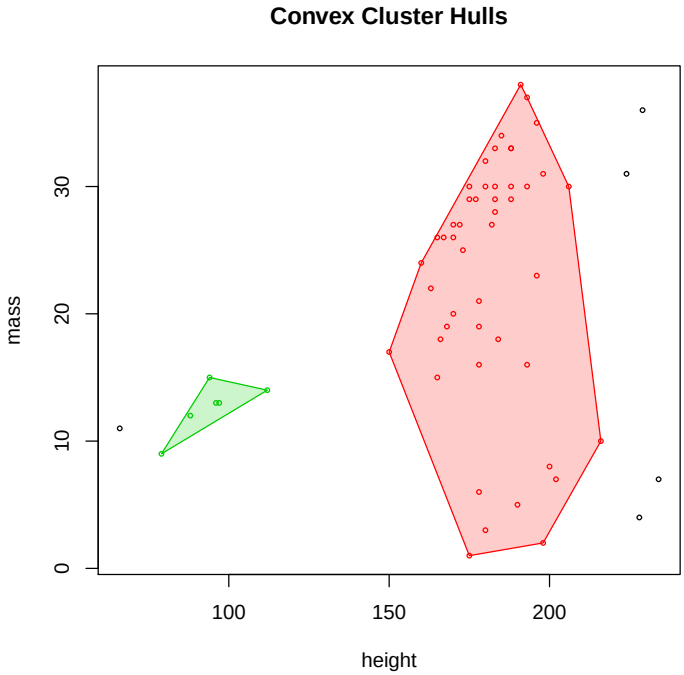
Una vez hemos instalado los paquetes necesarios, calcularemos cual es el valor de **eps** óptimo para nuestra clasificación DBSCAN. Para ello, utilizaremos el método k-nearest neighbour, y buscaremos donde exista un cambio drástico en nuestra gráfica:

```
> kNNdistplot(medidas, k=4)
> abline(h=16.5, col="red")
```



En este ejemplo podemos observar que el cambio se encuentra en el **valor 16.5**, por lo que ese será nuestro valor para eps. A continuación, realizaremos la clasificación dbscan y mostraremos una gráfica con los resultados:

```
> set.seed(123)
> clasificacionDBSCAN <- fpc::dbscan(medidas, eps = 16.5, MinPts = 5)
> hullplot(medidas, clasificacionDBSCAN$cluster)
```



Como podemos observar en el gráfico, el algoritmo nos ha separado los personajes en 2 clusters, un pequeño cluster de color **verde** que incluye 6 de estos, y un gran cluster **rojo** que incluye la gran mayoria, 47. Por último encontramos aquellos puntos negros que no están ni en el cluster verde ni en el rojo, estos puntos se corresponde con valores **outlier**.