Clustering: 2ª parte

Code

Breast Cancer Wisconsin (Diagnostic) Data Set

Vamos a trabajar con datos sobre un estudio realizado con datos sobre cáncer de mama. Entramos en el enlace de Kaggle: https://www.kaggle.com/uciml/breast-cancer-wisconsin-data (https://www.kaggle.com/uciml/breast-cancer-wisconsin-data) y leemos con detenimiento toda la información acerca de este *dataset*. Los datos originales están tomados de la UCI Repository: https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29 (https://archive.ics.uci.edu/ml/datasets/Breast+Cancer+Wisconsin+%28Diagnostic%29)

Exploración inicial

En primer lugar leemos la información sobre estos datos disponible en Kaggle: ¿qué tipo de estudio es?, ¿cómo se obtuvieron los datos?, ¿en qué características relevantes debemos fijarnos? En segundo lugar, importamos los datos y los estudiamos. ¿Cuántas variables o columnas del conjunto de datos son de tipo numérico? ¿Cuáles son sus nombres? ¿Hay alguna que tenga una etiqueta del tipo "enferma vs. no-enferma"?

Leemos los datos y observamos la estructura del dataframe.

Hide

```
# Importamos los datos y los estudiamos
```

```
There were 16 warnings (use warnings() to see them)
```

Hide

```
wisc_df<-read.csv("data/WisconsinCancer.csv")
#
str(wisc_df)
```

```
'data.frame':   569 obs. of  33 variables:
 $ id                     : int  842302 842517 84300903 84348301 84358402 843786 844359 84458
202 844981 84501001 ...
 $ diagnosis              : chr  "M" "M" "M" "M" ...
 $ radius_mean            : num  18 20.6 19.7 11.4 20.3 ...
 $ texture_mean           : num  10.4 17.8 21.2 20.4 14.3 ...
 $ perimeter_mean         : num  122.8 132.9 130 77.6 135.1 ...
 $ area_mean              : num  1001 1326 1203 386 1297 ...
 $ smoothness_mean        : num  0.1184 0.0847 0.1096 0.1425 0.1003 ...
 $ compactness_mean       : num  0.2776 0.0786 0.1599 0.2839 0.1328 ...
 $ concavity_mean         : num  0.3001 0.0869 0.1974 0.2414 0.198 ...
 $ concave.points_mean    : num  0.1471 0.0702 0.1279 0.1052 0.1043 ...
 $ symmetry_mean          : num  0.242 0.181 0.207 0.26 0.181 ...
 $ fractal_dimension_mean : num  0.0787 0.0567 0.06 0.0974 0.0588 ...
 $ radius_se              : num  1.095 0.543 0.746 0.496 0.757 ...
 $ texture_se             : num  0.905 0.734 0.787 1.156 0.781 ...
 $ perimeter_se           : num  8.59 3.4 4.58 3.44 5.44 ...
 $ area_se                : num  153.4 74.1 94 27.2 94.4 ...
 $ smoothness_se          : num  0.0064 0.00522 0.00615 0.00911 0.01149 ...
 $ compactness_se         : num  0.049 0.0131 0.0401 0.0746 0.0246 ...
 $ concavity_se           : num  0.0537 0.0186 0.0383 0.0566 0.0569 ...
 $ concave.points_se      : num  0.0159 0.0134 0.0206 0.0187 0.0188 ...
 $ symmetry_se            : num  0.03 0.0139 0.0225 0.0596 0.0176 ...
 $ fractal_dimension_se   : num  0.00619 0.00353 0.00457 0.00921 0.00511 ...
 $ radius_worst           : num  25.4 25 23.6 14.9 22.5 ...
 $ texture_worst          : num  17.3 23.4 25.5 26.5 16.7 ...
 $ perimeter_worst        : num  184.6 158.8 152.5 98.9 152.2 ...
 $ area_worst             : num  2019 1956 1709 568 1575 ...
 $ smoothness_worst       : num  0.162 0.124 0.144 0.21 0.137 ...
 $ compactness_worst      : num  0.666 0.187 0.424 0.866 0.205 ...
 $ concavity_worst        : num  0.712 0.242 0.45 0.687 0.4 ...
 $ concave.points_worst   : num  0.265 0.186 0.243 0.258 0.163 ...
 $ symmetry_worst         : num  0.46 0.275 0.361 0.664 0.236 ...
 $ fractal_dimension_worst: num  0.1189 0.089 0.0876 0.173 0.0768 ...
 $ X                      : logi  NA NA NA NA NA NA ...
```

Le echamos un vistazo a los datos.

Hide

```
head(wisc_df)
```

```
        id diagnosis radius_mean texture_mean perimeter_mean area_mean smoothness_mean compac
tness_mean concavity_mean
1   842302         M       17.99        10.38         122.80    1001.0         0.11840
0.27760         0.3001
2   842517         M       20.57        17.77         132.90    1326.0         0.08474
0.07864         0.0869
3 84300903         M       19.69        21.25         130.00    1203.0         0.10960
0.15990         0.1974
4 84348301         M       11.42        20.38          77.58     386.1         0.14250
0.28390         0.2414
5 84358402         M       20.29        14.34         135.10    1297.0         0.10030
0.13280         0.1980
6   843786         M       12.45        15.70          82.57     477.1         0.12780
0.17000         0.1578
  concave.points_mean symmetry_mean fractal_dimension_mean radius_se texture_se perimeter_se
area_se smoothness_se
1             0.14710        0.2419                0.07871    1.0950     0.9053        8.589
153.40      0.006399
2             0.07017        0.1812                0.05667    0.5435     0.7339        3.398
74.08       0.005225
3             0.12790        0.2069                0.05999    0.7456     0.7869        4.585
94.03       0.006150
4             0.10520        0.2597                0.09744    0.4956     1.1560        3.445
27.23       0.009110
5             0.10430        0.1809                0.05883    0.7572     0.7813        5.438
94.44       0.011490
6             0.08089        0.2087                0.07613    0.3345     0.8902        2.217
27.19       0.007510
  compactness_se concavity_se concave.points_se symmetry_se fractal_dimension_se radius_worst
texture_worst
1        0.04904      0.05373           0.01587     0.03003             0.006193        25.38
17.33
2        0.01308      0.01860           0.01340     0.01389             0.003532        24.99
23.41
3        0.04006      0.03832           0.02058     0.02250             0.004571        23.57
25.53
4        0.07458      0.05661           0.01867     0.05963             0.009208        14.91
26.50
5        0.02461      0.05688           0.01885     0.01756             0.005115        22.54
16.67
6        0.03345      0.03672           0.01137     0.02165             0.005082        15.47
23.75
  perimeter_worst area_worst smoothness_worst compactness_worst concavity_worst concave.point
s_worst symmetry_worst
1          184.60     2019.0           0.1622            0.6656          0.7119
0.2654          0.4601
2          158.80     1956.0           0.1238            0.1866          0.2416
0.1860          0.2750
3          152.50     1709.0           0.1444            0.4245          0.4504
0.2430          0.3613
4           98.87      567.7           0.2098            0.8663          0.6869
0.2575          0.6638
5          152.20     1575.0           0.1374            0.2050          0.4000
0.1625          0.2364
6          103.40      741.6           0.1791            0.5249          0.5355
```

```
0.1741         0.3985
  fractal_dimension_worst  X
1                 0.11890 NA
2                 0.08902 NA
3                 0.08758 NA
4                 0.17300 NA
5                 0.07678 NA
6                 0.12440 NA
```

Para hacer un análisis de clustering nos quedamos solo con la parte numérica. Es decir, construimos una matriz con las variables con valores numéricos. Es decir, todas las columnas menos la primera columna, el *id* y la segunda *diagnosis*. No consideramos tampoco la última columna que no aporta información. Nombre con la información de los *id* a las filas de dicha matriz.

Hide

```
# Construimos una matriz: desde la columna 3 hasta la 32
wisc_data<-as.matrix(wisc_df[3:32])
# Le ponemos los ids a las filas de la matriz
rownames(wisc_data)<-wisc_df$id
```

¿Están todos los datos en el mismo rango de valores? Si no fuera así, debemos escalarlos o poner todas las variables en el mismo rango de valores.

Nota: en general se debe estudiar la media y la desviación para ver si se debe normalizar o no los datos: *colMeans(wisc_data)* y *apply(wisc_data,2,sd)*

Hide

```
# ¿Escalo los datos?: Lo vamos a hacer al aplicar el clustering
summary(wisc_data)
```

```
  radius_mean      texture_mean   perimeter_mean     area_mean      smoothness_mean   compact
ness_mean
 Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 143.5   Min.   :0.05263   Min.
:0.01938
 1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17   1st Qu.: 420.3   1st Qu.:0.08637   1st Q
u.:0.06492
 Median :13.370   Median :18.84   Median : 86.24   Median : 551.1   Median :0.09587   Median
:0.09263
 Mean   :14.127   Mean   :19.29   Mean   : 91.97   Mean   : 654.9   Mean   :0.09636   Mean
:0.10434
 3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10   3rd Qu.: 782.7   3rd Qu.:0.10530   3rd Q
u.:0.13040
 Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :2501.0   Max.   :0.16340   Max.
:0.34540
 concavity_mean    concave.points_mean symmetry_mean    fractal_dimension_mean   radius_se
texture_se
 Min.   :0.00000   Min.   :0.00000     Min.   :0.1060   Min.   :0.04996        Min.   :0.1115
Min.   :0.3602
 1st Qu.:0.02956   1st Qu.:0.02031     1st Qu.:0.1619   1st Qu.:0.05770        1st Qu.:0.2324
1st Qu.:0.8339
 Median :0.06154   Median :0.03350     Median :0.1792   Median :0.06154        Median :0.3242
Median :1.1080
 Mean   :0.08880   Mean   :0.04892     Mean   :0.1812   Mean   :0.06280        Mean   :0.4052
Mean   :1.2169
 3rd Qu.:0.13070   3rd Qu.:0.07400     3rd Qu.:0.1957   3rd Qu.:0.06612        3rd Qu.:0.4789
3rd Qu.:1.4740
 Max.   :0.42680   Max.   :0.20120     Max.   :0.3040   Max.   :0.09744        Max.   :2.8730
Max.   :4.8850
  perimeter_se       area_se        smoothness_se      compactness_se      concavity_se     c
oncave.points_se
 Min.   : 0.757   Min.   :  6.802   Min.   :0.001713   Min.   :0.002252   Min.   :0.00000   M
in.   :0.000000
 1st Qu.: 1.606   1st Qu.: 17.850   1st Qu.:0.005169   1st Qu.:0.013080   1st Qu.:0.01509   1
st Qu.:0.007638
 Median : 2.287   Median : 24.530   Median :0.006380   Median :0.020450   Median :0.02589   M
edian :0.010930
 Mean   : 2.866   Mean   : 40.337   Mean   :0.007041   Mean   :0.025478   Mean   :0.03189   M
ean   :0.011796
 3rd Qu.: 3.357   3rd Qu.: 45.190   3rd Qu.:0.008146   3rd Qu.:0.032450   3rd Qu.:0.04205   3
rd Qu.:0.014710
 Max.   :21.980   Max.   :542.200   Max.   :0.031130   Max.   :0.135400   Max.   :0.39600   M
ax.   :0.052790
  symmetry_se       fractal_dimension_se  radius_worst   texture_worst   perimeter_worst    a
rea_worst
 Min.   :0.007882   Min.   :0.0008948    Min.   : 7.93   Min.   :12.02   Min.   : 50.41   Mi
n.   : 185.2
 1st Qu.:0.015160   1st Qu.:0.0022480    1st Qu.:13.01   1st Qu.:21.08   1st Qu.: 84.11   1st
Qu.: 515.3
 Median :0.018730   Median :0.0031870    Median :14.97   Median :25.41   Median : 97.66   Med
ian : 686.5
 Mean   :0.020542   Mean   :0.0037949    Mean   :16.27   Mean   :25.68   Mean   :107.26   Mea
n   : 880.6
 3rd Qu.:0.023480   3rd Qu.:0.0045580    3rd Qu.:18.79   3rd Qu.:29.72   3rd Qu.:125.40   3rd
Qu.:1084.0
 Max.   :0.078950   Max.   :0.0298400    Max.   :36.04   Max.   :49.54   Max.   :251.20   Ma
```

```
x.   :4254.0
 smoothness_worst  compactness_worst concavity_worst  concave.points_worst symmetry_worst   f
ractal_dimension_worst
 Min.   :0.07117   Min.   :0.02729   Min.   :0.0000   Min.   :0.00000      Min.   :0.1565   M
in.   :0.05504
 1st Qu.:0.11660   1st Qu.:0.14720   1st Qu.:0.1145   1st Qu.:0.06493      1st Qu.:0.2504   1
st Qu.:0.07146
 Median :0.13130   Median :0.21190   Median :0.2267   Median :0.09993      Median :0.2822   M
edian :0.08004
 Mean   :0.13237   Mean   :0.25427   Mean   :0.2722   Mean   :0.11461      Mean   :0.2901   M
ean   :0.08395
 3rd Qu.:0.14600   3rd Qu.:0.33910   3rd Qu.:0.3829   3rd Qu.:0.16140      3rd Qu.:0.3179   3
rd Qu.:0.09208
 Max.   :0.22260   Max.   :1.05800   Max.   :1.2520   Max.   :0.29100      Max.   :0.6638   M
ax.   :0.20750
```

Observamos que no es así, por lo que los escalamos.

Hide

```
wisc_data_escalados<-scale(wisc_data)
summary(wisc_data_escalados)
```

```
  radius_mean       texture_mean     perimeter_mean      area_mean       smoothness_mean    c
ompactness_mean
 Min.   :-2.0279   Min.   :-2.2273   Min.   :-1.9828   Min.   :-1.4532   Min.   :-3.10935   M
in.   :-1.6087
 1st Qu.:-0.6888   1st Qu.:-0.7253   1st Qu.:-0.6913   1st Qu.:-0.6666   1st Qu.:-0.71034   1
st Qu.:-0.7464
 Median :-0.2149   Median :-0.1045   Median :-0.2358   Median :-0.2949   Median :-0.03486   M
edian :-0.2217
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.00000   M
ean   : 0.0000
 3rd Qu.: 0.4690   3rd Qu.: 0.5837   3rd Qu.: 0.4992   3rd Qu.: 0.3632   3rd Qu.: 0.63564   3
rd Qu.: 0.4934
 Max.   : 3.9678   Max.   : 4.6478   Max.   : 3.9726   Max.   : 5.2459   Max.   : 4.76672   M
ax.   : 4.5644
 concavity_mean    concave.points_mean symmetry_mean      fractal_dimension_mean   radius_se
texture_se
 Min.   :-1.1139   Min.   :-1.2607     Min.   :-2.74171   Min.   :-1.8183        Min.   :-1.0
590   Min.   :-1.5529
 1st Qu.:-0.7431   1st Qu.:-0.7373     1st Qu.:-0.70262   1st Qu.:-0.7220        1st Qu.:-0.6
230   1st Qu.:-0.6942
 Median :-0.3419   Median :-0.3974     Median :-0.07156   Median :-0.1781        Median :-0.2
920   Median :-0.1973
 Mean   : 0.0000   Mean   : 0.0000     Mean   : 0.00000   Mean   : 0.0000        Mean   : 0.0
000   Mean   : 0.0000
 3rd Qu.: 0.5256   3rd Qu.: 0.6464     3rd Qu.: 0.53031   3rd Qu.: 0.4706        3rd Qu.: 0.2
659   3rd Qu.: 0.4661
 Max.   : 4.2399   Max.   : 3.9245     Max.   : 4.48081   Max.   : 4.9066        Max.   : 8.8
991   Max.   : 6.6494
  perimeter_se        area_se        smoothness_se     compactness_se     concavity_se     co
ncave.points_se
 Min.   :-1.0431   Min.   :-0.7372   Min.   :-1.7745   Min.   :-1.2970   Min.   :-1.0566   Mi
n.   :-1.9118
 1st Qu.:-0.6232   1st Qu.:-0.4943   1st Qu.:-0.6235   1st Qu.:-0.6923   1st Qu.:-0.5567   1s
t Qu.:-0.6739
 Median :-0.2864   Median :-0.3475   Median :-0.2201   Median :-0.2808   Median :-0.1989   Me
dian :-0.1404
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Me
an   : 0.0000
 3rd Qu.: 0.2428   3rd Qu.: 0.1067   3rd Qu.: 0.3680   3rd Qu.: 0.3893   3rd Qu.: 0.3365   3r
d Qu.: 0.4722
 Max.   : 9.4537   Max.   :11.0321   Max.   : 8.0229   Max.   : 6.1381   Max.   :12.0621   Ma
x.   : 6.6438
  symmetry_se      fractal_dimension_se  radius_worst     texture_worst      perimeter_worst
area_worst
 Min.   :-1.5315   Min.   :-1.0960      Min.   :-1.7254   Min.   :-2.22204   Min.   :-1.6919
Min.   :-1.2213
 1st Qu.:-0.6511   1st Qu.:-0.5846      1st Qu.:-0.6743   1st Qu.:-0.74797   1st Qu.:-0.6890
1st Qu.:-0.6416
 Median :-0.2192   Median :-0.2297      Median :-0.2688   Median :-0.04348   Median :-0.2857
Median :-0.3409
 Mean   : 0.0000   Mean   : 0.0000      Mean   : 0.0000   Mean   : 0.00000   Mean   : 0.0000
Mean   : 0.0000
 3rd Qu.: 0.3554   3rd Qu.: 0.2884      3rd Qu.: 0.5216   3rd Qu.: 0.65776   3rd Qu.: 0.5398
3rd Qu.: 0.3573
 Max.   : 7.0657   Max.   : 9.8429      Max.   : 4.0906   Max.   : 3.88249   Max.   : 4.2836
```

```
Max.   : 5.9250
 smoothness_worst  compactness_worst concavity_worst   concave.points_worst symmetry_worst
fractal_dimension_worst
 Min.   :-2.6803   Min.   :-1.4426   Min.   :-1.3047   Min.   :-1.7435      Min.   :-2.1591
Min.   :-1.6004
 1st Qu.:-0.6906   1st Qu.:-0.6805   1st Qu.:-0.7558   1st Qu.:-0.7557      1st Qu.:-0.6413
1st Qu.:-0.6913
 Median :-0.0468   Median :-0.2693   Median :-0.2180   Median :-0.2233      Median :-0.1273
Median :-0.2163
 Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000   Mean   : 0.0000      Mean   : 0.0000
Mean   : 0.0000
 3rd Qu.: 0.5970   3rd Qu.: 0.5392   3rd Qu.: 0.5307   3rd Qu.: 0.7119      3rd Qu.: 0.4497
3rd Qu.: 0.4504
 Max.   : 3.9519   Max.   : 5.1084   Max.   : 4.6965   Max.   : 2.6835      Max.   : 6.0407
Max.   : 6.8408
```

Antes de continuar nos planteamos si es conveniente reducir el número de variables aplicando una técnica de redución de la dimensionalidad.

Hide

```
dim(wisc_data_escalados)
```

```
[1] 569  30
```

Los datos tienen 30 variables o atributos. Decidimos que vamos a reducir la dimensionalidad.

Análisis de Componentes Principales o **PCA**

PCA o Análisis de Componentes Principales sirve para realizar una trasformación de los datos de tal manera que se trabaje con menos variables y sea más sencillo aplicar los algoritmos. Es una técnica de "reducción de la dimensionalidad". Enlace: https://bit.ly/145NDZW (https://bit.ly/145NDZW)

Hide

```
#Aplicamos un PCA escalando los datos
wisc_pca<-prcomp(wisc_data_escalados,scale. = TRUE)
```

Nota: el parámetro "scale" está por defecto a false. En general se suele hacer la llamada poniéndolo a true. En este caso, al haber escalado los datos antes, no sería necesario. (Cuando hemos escalado los datos no sabíamos si íbamos a aplicar o no un PCA).

Aplicamos la función *summary* para ver con cuántos ejes se obtienen con, por ejemplo, un 90% de "variabilidad" de los datos. Observamos el valor de *cumulative proportion*. ¿Con cuántos ejes se consigue un 90% de variablidad?

Nota: el objetivo es tomar un número de componentes del PCA menor que el número oringinal de variables y tal que la variabilidad de los datos, su "cumulative proportion", sea mayor que 0.9.

Un PCA lo que hace es representar los datos en un nuevo sistema de coordenadas de tal forma que los datos se ordenan según su "variabilidad". Eliminamos variables - para trabajar en un espacio métrico más sencillo -, pero de manera que no perdamos demasiada información.

Hide

```
# ...en este caso para PC7 (ver cumulative proportion)
# ...con 7 ejes se consigue un 90% de variabilidad de los datos
summary(wisc_pca)
```

```
Importance of components:
                          PC1    PC2     PC3     PC4     PC5     PC6     PC7     PC8    PC9
PC10   PC11    PC12
Standard deviation     3.6444 2.3857 1.67867 1.40735 1.28403 1.09880 0.82172 0.69037 0.6457
0.59219 0.5421 0.51104
Proportion of Variance 0.4427 0.1897 0.09393 0.06602 0.05496 0.04025 0.02251 0.01589 0.0139
0.01169 0.0098 0.00871
Cumulative Proportion  0.4427 0.6324 0.72636 0.79239 0.84734 0.88759 0.91010 0.92598 0.9399
0.95157 0.9614 0.97007
                          PC13    PC14    PC15    PC16    PC17    PC18    PC19    PC20   PC21
PC22    PC23   PC24
Standard deviation     0.49128 0.39624 0.30681 0.28260 0.24372 0.22939 0.22244 0.17652 0.1731
0.16565 0.15602 0.1344
Proportion of Variance 0.00805 0.00523 0.00314 0.00266 0.00198 0.00175 0.00165 0.00104 0.0010
0.00091 0.00081 0.0006
Cumulative Proportion  0.97812 0.98335 0.98649 0.98915 0.99113 0.99288 0.99453 0.99557 0.9966
0.99749 0.99830 0.9989
                          PC25    PC26    PC27    PC28    PC29    PC30
Standard deviation     0.12442 0.09043 0.08307 0.03987 0.02736 0.01153
Proportion of Variance 0.00052 0.00027 0.00023 0.00005 0.00002 0.00000
Cumulative Proportion  0.99942 0.99969 0.99992 0.99997 1.00000 1.00000
```

Observamos que con 7 ejes se consigue un 90% de variabilidad de los datos. Es decir, nos quedamos con las primeras siete componentes.

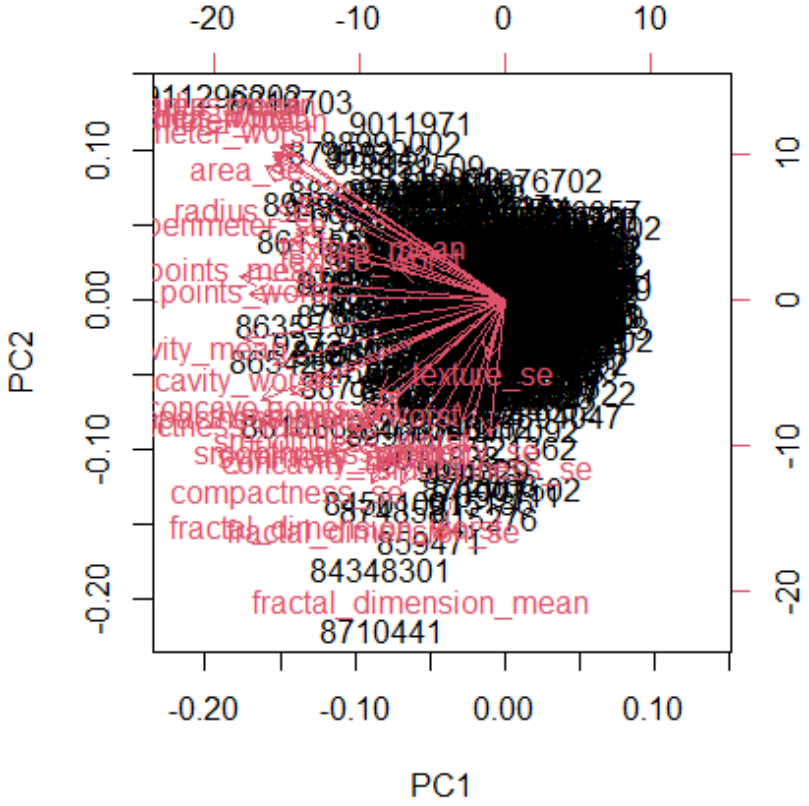
Hide

```
wisc_data_reduccion<-wisc_pca$x[,1:7]
```

Aunque no sea necesario, visualizamos el resultado del PCA de tal manera que vea la relación de las variables originales.

Hide

```
# Completo: indica la relacion entre los ejes originales
biplot(wisc_pca)
```



Para tratar de entender qué significa hacer un PCA representamos dos a dos algunos ejes. ¿Qué dibujo separa mejor? ¿PC1 vs. PC3? o ¿PC1 vs. PC9?

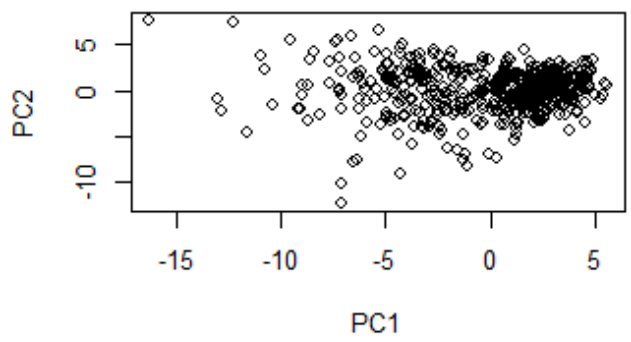
Representamos los ejes dos a dos. Por ejemplo, PC1 vs. PC2 o PC1 vs. PC9 y observémoslos según la etiqueta. ¿Qué dibujo separa mejor? ¿PC1 vs. PC3? o ¿PC1 vs. PC9?

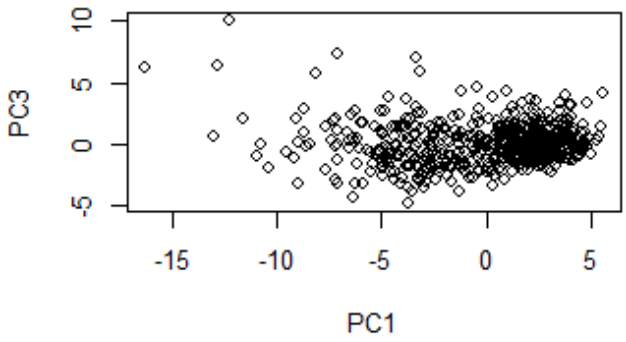
Hide

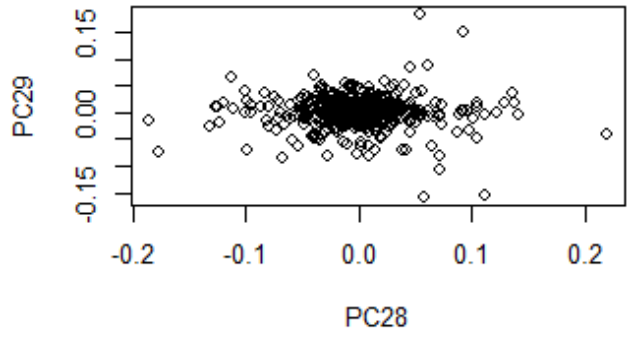
```
par(mfrow = c(2,2))
plot(wisc_pca$x[,c(1,2)],xlab = "PC1",ylab = "PC2")
plot(wisc_pca$x[,c(1,3)],xlab = "PC1",ylab = "PC3")
```

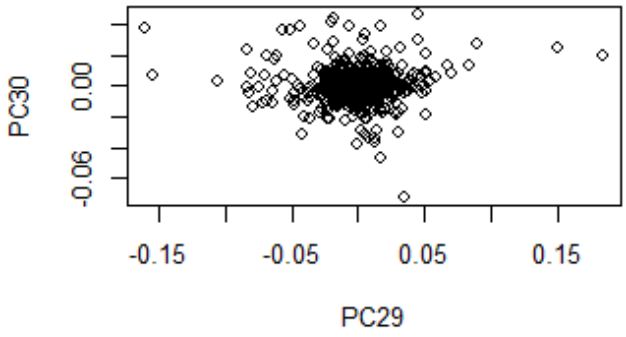
Hide

```
plot(wisc_pca$x[,c(28,29)],xlab = "PC28",ylab = "PC29")
plot(wisc_pca$x[,c(29,30)],xlab = "PC29",ylab = "PC30")
```









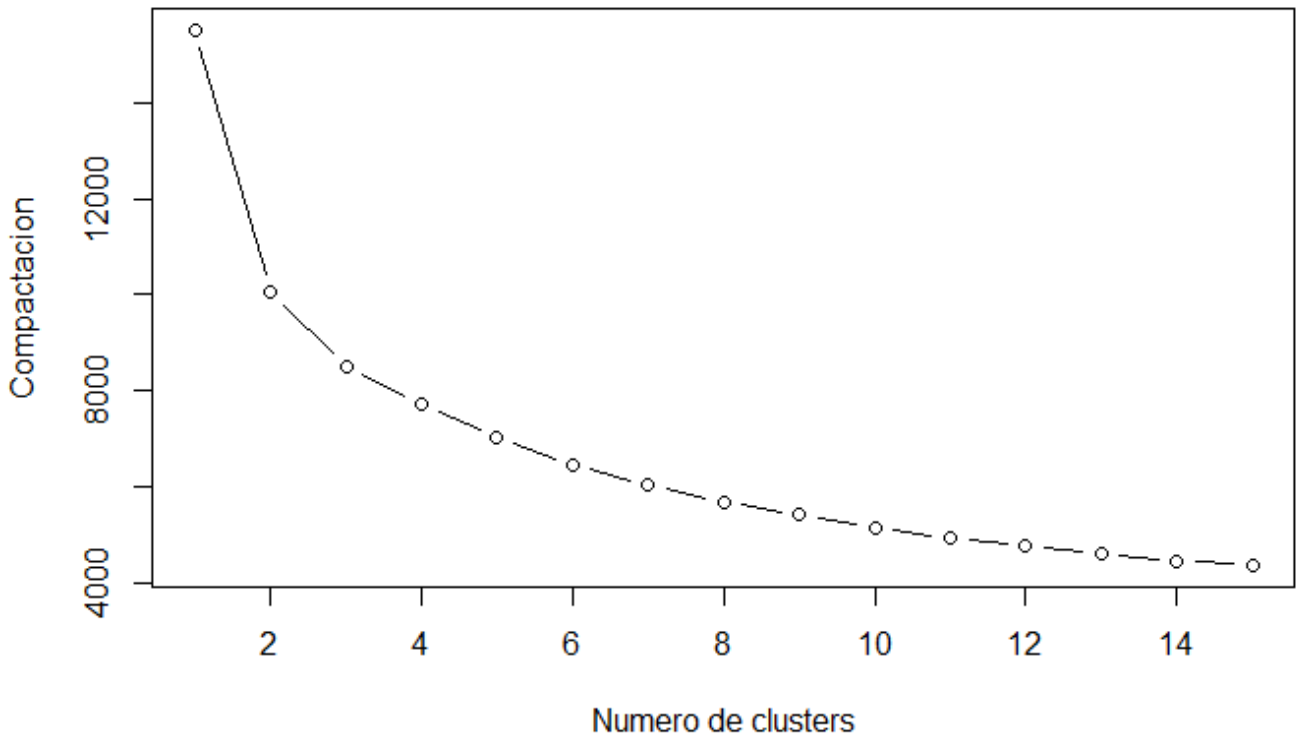
Clustering k-Means

Aplicamos el algoritmo k-Means.

Debemos tener en cuenta la elección del valor de k y eligir un valor de los parámetros de tal manera que se controle la aleatoriedad intrínseca del algoritmo. Además, siempre se deben escalar los datos para ejecutar el algoritmo.

Hide

```
vector_compactacion<-0
for(i in 1:15){
  km_wisc_data_reduccion<-kmeans(wisc_data_reduccion,center=i,nstar=20)
  vector_compactacion[i] <- km_wisc_data_reduccion$tot.withinss
}
# Construye rejilla 1x1
par(mfrow = c(1,1))
# Representamos sum of squares vs. number of clusters
plot(1:15, vector_compactacion, type = "b",
     xlab = "Numero de clusters",
     ylab = "Compactacion")
```



Observando la gráfica se puede elegir un valor de k o bien 2, 3 o incluso 4. No está claro del todo. Realizamos ejecuciones para k=2 y k=4.

Aplicamos el algoritmo kMeans para k=2.

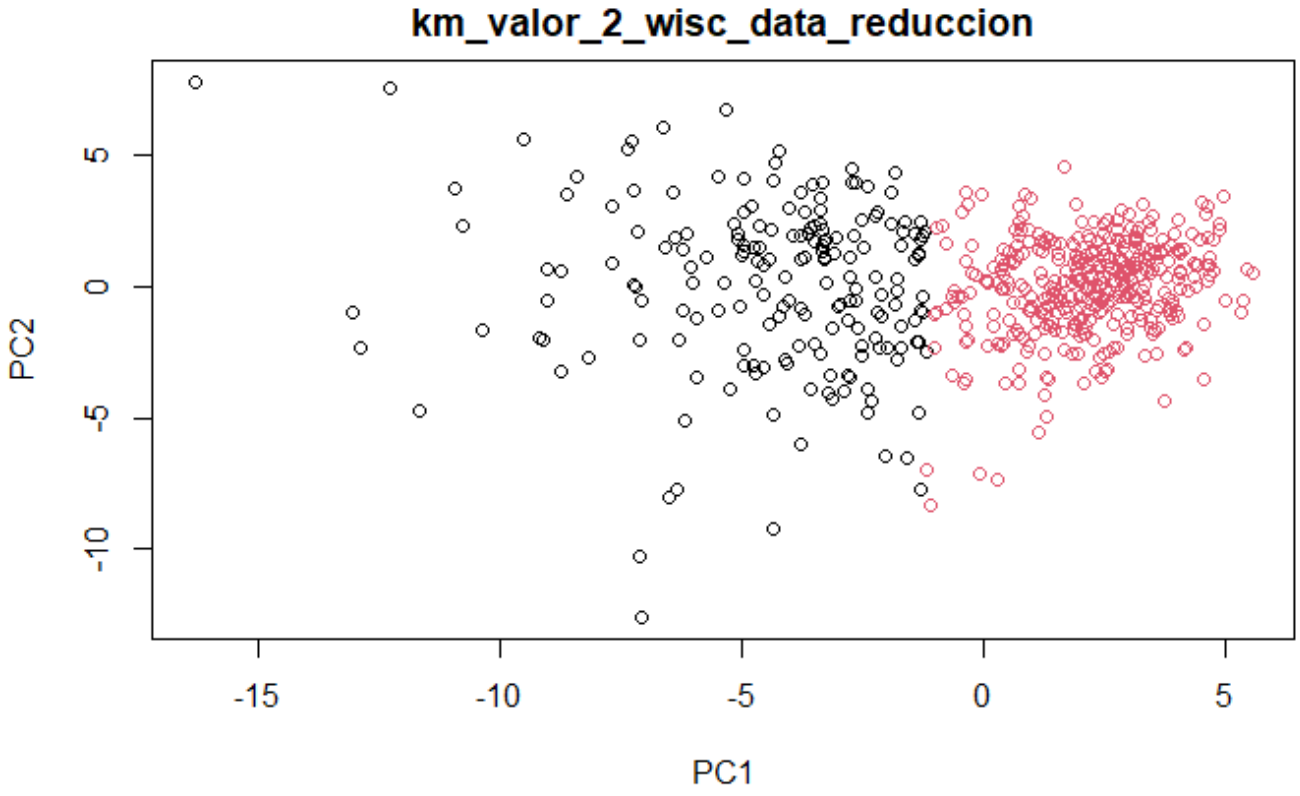
Hide

```
km_valor_2_wisc_data_reduccion<-kmeans(wisc_data_reduccion,center=2,nstar=20)
#km_valor_2_wisc_data_reduccion
```

Visualizamos el resultado (la gráfica visualiza los dos primeros ejes de los siete que tienen los datos reducidos).

Hide

```
plot(wisc_data_reduccion,col=km_valor_2_wisc_data_reduccion$cluster, main="km_valor_2_wisc_da
ta_reduccion")
```



Aplicamos el algoritmo kMeans para k=4.

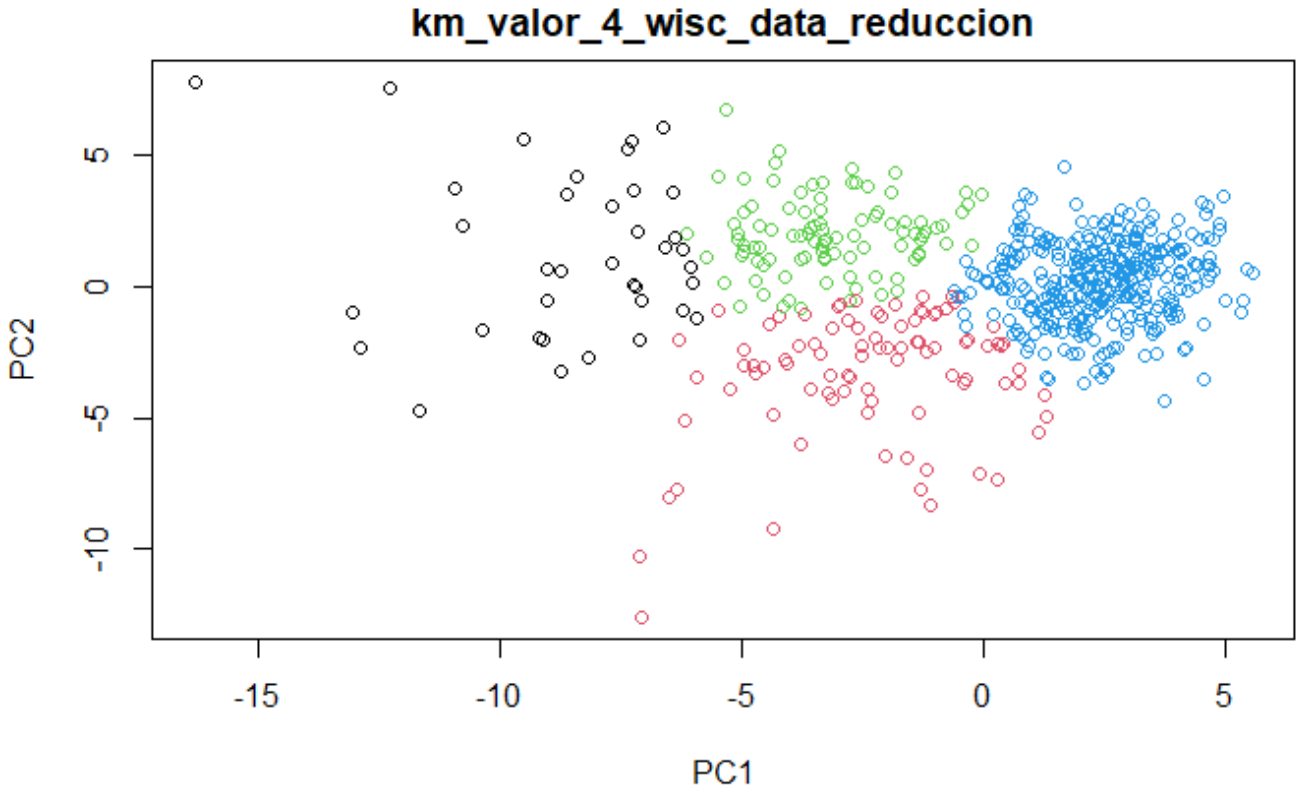
Hide

```
km_valor_4_wisc_data_reduccion<-kmeans(wisc_data_reduccion,center=4,nstar=20)
#km_valor_4_wisc_data_reduccion
```

Visualizamos el resultado (la gráfica visualiza los dos primeros ejes de los siete que tienen los datos reducidos).

Hide

```
plot(wisc_data_reduccion,col=km_valor_4_wisc_data_reduccion$cluster, main="km_valor_4_wisc_da
ta_reduccion")
```



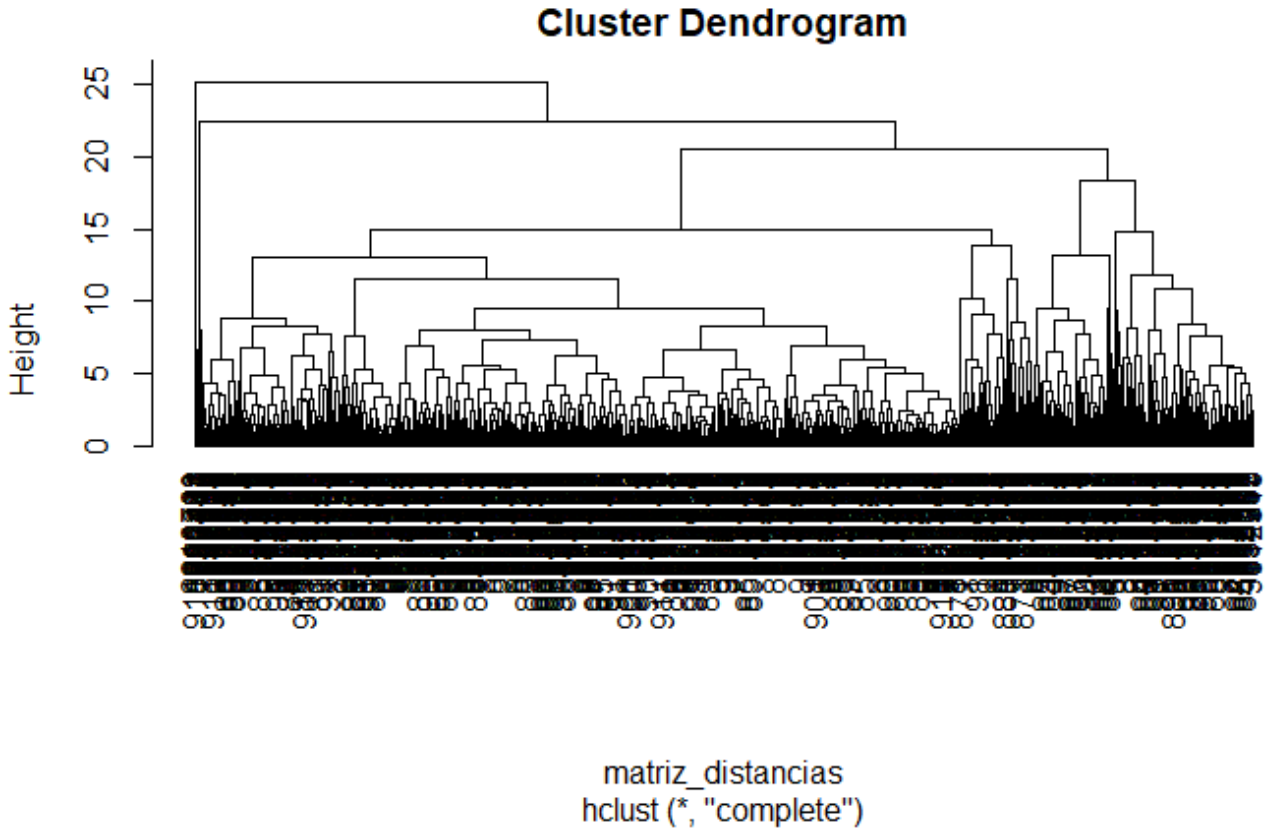
Clustering jerárquico

Realizamos un clustering de tipo jerárquico, para ello tenemos en cuenta las siguientes consideraciones:

- Los datos siempre tienen que estar escalados para que todas las observaciones estén en un rango parecido.
- Se calcula la matriz de distancias.
- Vamos a utilizar la opción de "linkage Completo"
- Se construye el dendograma y generamos los clústeres.

Hide

```
# Calculo de la matriz de distancias
matriz_distancias<-dist(wisc_data_reduccion)
# Clustering jerarquico con linkage Completo
hclust_wisc_data_reduccion<-hclust(matriz_distancias,method = "complete")
# Visualizamos dendograma
plot(hclust_wisc_data_reduccion, hang=-1)
```



Observando el dendograma vemos que no tiene sentido generar un resultado con dos clústeres. Lo cortamos de maneras que obtengamos cuatro.

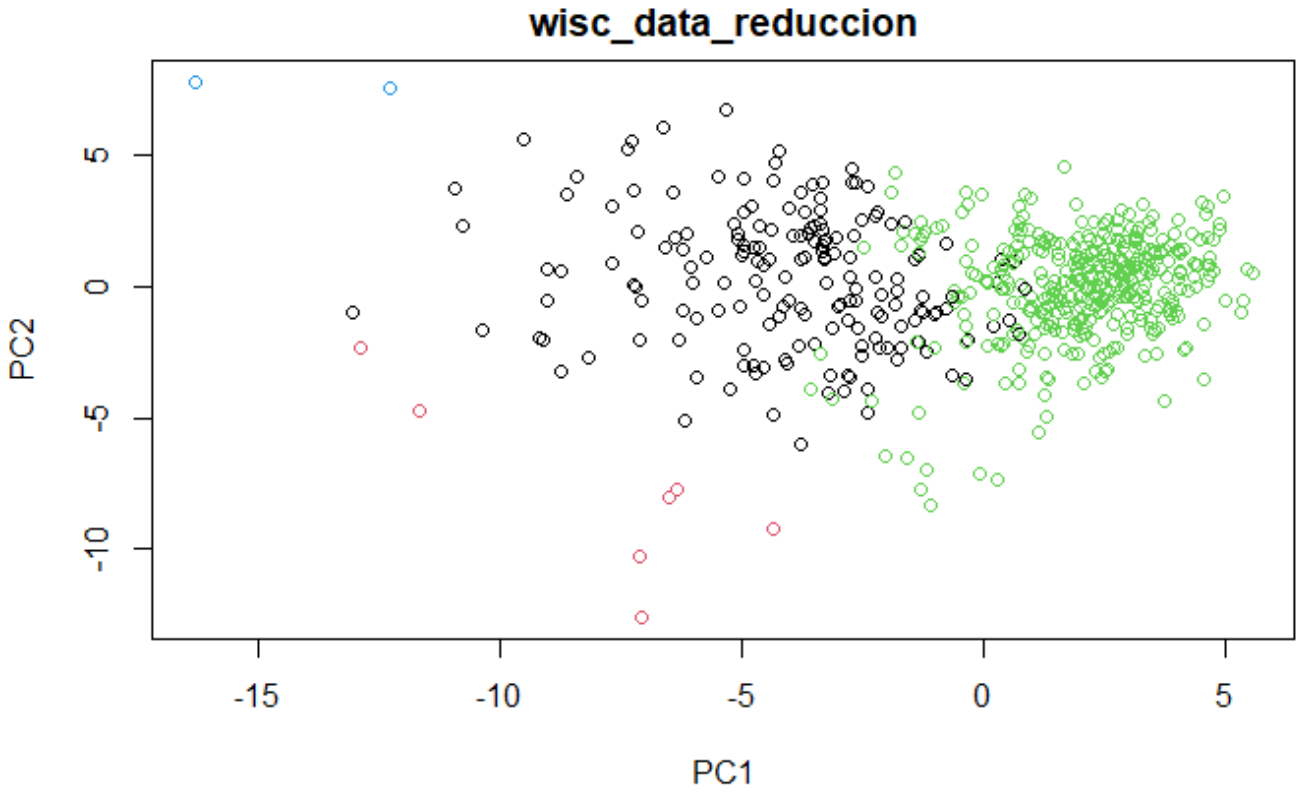
Hide

```
# Cortamos de manera que tengamos 4 clusteres
wisc_hclust_clusters<-cutree(wisc_hclust,k=4)
```

Visualizamos el resultado (la gráfica visualiza los dos primeros ejes de los siete que tienen los datos reducidos).

Hide

```
plot(wisc_data_reduccion,col=wisc_hclust_clusters, main="wisc_data_reduccion")
```



Comparamos los resultados de los dos algoritmos.

Primero comparamos la opción del kMeans para k=2. ¿Qué observamos?

Hide

```
table(km_valor_2_wisc_data_reduccion$cluster, wisc_hclust_clusters)
```

```
   wisc_hclust_clusters
      1   2   3   4
  1 160   7  20   2
  2  17   0 363   0
```

Segundo comparamos la opción del kMeans para k=4. ¿Qué observamos?

Hide

```
table(km_valor_4_wisc_data_reduccion$cluster, wisc_hclust_clusters)
```

```
   wisc_hclust_clusters
      1   2   3   4
  1  33   2   0   2
  2  58   5  27   0
  3  79   0  18   0
  4   7   0 338   0
```

Conocimiento experto del problema o **the ground truth**

El conjunto de datos que estamos analizando incluye una etiqueta que informa si el paciente está o no enfermo. Esta etiqueta se puede considerar el conocimiento experto o **the ground truth** del problema. En general, este tipo de información adicional no estará disponible en los primeros pasos de un estudio. De

hecho, muchas veces, el realizar un estudio de clustering sirve para o bien definir las etiquetas o bien para ver la calidad y coherencia de las mismas.

Por ejemplo, podríamos haber elegido directamente k=2 al ejecutar el algoritmo del kMeans, pero estaríamos asumiendo como hipótesis que, por tener dos clases, los datos se agrupan en dos grupos. La asunción de esta hipótesis es un salto al vacío y desvirtúa el análisis de clustering. Si queremos utilizar un algoritmo de clustering para ver los grupos que hay y, como hipótesis asumimos que los grupos vienen determinados de manera externa a través de una etiqueta, ¿qué sentido tiene el planteamiento del estudio? En general, las etiquetas están bien construidas y se puede asumir la hipótesis, pero no deja de ser una decisión ajena al estudio no supervisado de los datos.

Volviendo al problema, alamecenamos el vector de *diagnosis* como un vector binario de 0s y 1s.

Hide

```
# Guarda el vector de diagnostico
diagnosis<-wisc_df$diagnosis
diagnosis # observamos
```

```
  [1] "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "M" "B" "B" "B"
"M" "M" "M" "M" "M" "M" "M"
 [30] "M" "M" "M" "M" "M" "M" "M" "M" "B" "M" "M" "M" "M" "M" "M" "M" "M" "M" "B" "M" "B" "B" "B"
"B" "B" "M" "M" "B" "M" "M"
 [59] "B" "B" "B" "B" "M" "B" "M" "M" "B" "B" "B" "B" "M" "B" "M" "M" "B" "M" "B" "M" "M" "B"
"B" "B" "M" "M" "B" "M" "M"
 [88] "M" "B" "B" "B" "M" "B" "B" "M" "M" "B" "B" "B" "M" "M" "B" "B" "B" "B" "M" "B" "B" "M"
"B" "B" "B" "B" "B" "B" "B"
[117] "B" "M" "M" "M" "B" "M" "M" "B" "B" "B" "M" "M" "B" "M" "B" "M" "M" "B" "M" "M" "B" "B"
"M" "B" "B" "M" "B" "B" "B"
[146] "B" "M" "B" "B" "B" "B" "B" "B" "B" "B" "B" "M" "B" "B" "B" "B" "M" "M" "B" "M" "B" "B"
"M" "M" "B" "B" "M" "M" "B"
[175] "B" "B" "B" "M" "B" "B" "M" "M" "M" "B" "M" "B" "M" "B" "B" "B" "M" "B" "B" "M" "M" "B"
"M" "M" "M" "M" "B" "M" "M"
[204] "M" "B" "M" "B" "M" "B" "B" "M" "B" "M" "M" "M" "M" "B" "B" "M" "M" "B" "B" "B" "M" "B"
"B" "B" "B" "B" "M" "M" "B"
[233] "B" "M" "B" "B" "M" "M" "B" "M" "B" "B" "B" "B" "M" "B" "B" "B" "B" "B" "M" "B" "M" "M"
"M" "M" "M" "M" "M" "M" "M"
[262] "M" "M" "M" "M" "M" "B" "B" "B" "B" "B" "B" "M" "B" "M" "B" "B" "M" "B" "B" "M" "B" "M"
"M" "B" "B" "B" "B" "B" "B"
[291] "B" "B" "B" "B" "B" "B" "B" "M" "B" "B" "M" "B" "M" "B" "B" "B" "B" "B" "B" "B" "B" "B"
"B" "B" "B" "B" "B" "M" "B"
[320] "B" "B" "M" "B" "M" "B" "B" "B" "B" "M" "M" "M" "B" "B" "B" "B" "M" "B" "M" "B" "M" "B"
"B" "B" "M" "B" "B" "B" "B"
[349] "B" "B" "B" "M" "M" "M" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "M" "M" "B" "M" "M"
"M" "B" "M" "M" "B" "B" "B"
[378] "B" "B" "M" "B" "B" "B" "B" "B" "M" "B" "B" "B" "M" "B" "B" "M" "M" "B" "B" "B" "B" "B"
"B" "M" "B" "B" "B" "B" "B"
[407] "B" "B" "M" "B" "B" "B" "B" "B" "M" "B" "B" "M" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B"
"B" "B" "M" "B" "M" "M" "B"
[436] "M" "B" "B" "B" "B" "B" "M" "B" "B" "M" "B" "M" "B" "B" "M" "B" "M" "B" "B" "B" "B" "B"
"B" "B" "B" "M" "M" "B" "B"
[465] "B" "B" "B" "B" "M" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "M" "B" "B" "B" "B" "B" "B"
"B" "M" "B" "M" "B" "B" "M"
[494] "B" "B" "B" "B" "B" "M" "M" "B" "M" "B" "M" "B" "B" "B" "B" "B" "M" "B" "B" "M" "B" "M"
"B" "M" "M" "B" "B" "B" "M"
[523] "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "M" "B" "M" "M" "B" "B" "B" "B" "B" "B" "B"
"B" "B" "B" "B" "B" "B" "B"
[552] "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "B" "M" "M" "M" "M" "M" "M" "B"
```

Ponemos el vector como binario para poder utilizarlo (por ejemplo, para hacer un dibujo).

Hide

```
# Pone el vector en binario
diagnosis<-as.numeric(wisc_df$diagnosis == 'M')
diagnosis
```

```
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 1 1 1 1 1 1
1 1 0 1 0 0 0 0 0 1 1 0 1 1
 [59] 0 0 0 0 1 0 1 1 0 0 0 0 1 0 1 1 0 1 0 1 1 0 0 0 1 1 0 1 1 1 0 0 0 1 0 0 1 1 0 0 0 1 1 0
0 0 0 1 0 0 1 0 0 0 0 0 0 0
[117] 0 1 1 1 0 1 1 0 0 0 1 1 0 1 0 1 1 0 1 1 0 0 1 0 0 1 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0
0 1 1 0 1 0 0 1 1 0 0 1 1 0
[175] 0 0 0 1 0 0 1 1 1 0 1 0 1 0 0 0 1 0 0 1 1 0 1 1 1 1 0 1 1 1 0 1 0 1 0 0 1 0 1 1 1 1 0 0
1 1 0 0 0 1 0 0 0 0 0 1 1 0
[233] 0 1 0 0 1 1 0 1 0 0 0 0 1 0 0 0 0 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 0 0 0 0 1 0 1 0
0 1 0 0 1 0 1 1 0 0 0 0 0 0
[291] 0 0 0 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 1 0 1 0 0 0 0 1 1 1 0 0 0
0 1 0 1 0 1 0 0 0 1 0 0 0 0
[349] 0 0 0 1 1 1 0 0 0 0 0 0 0 0 0 0 1 1 0 1 1 1 0 1 1 0 0 0 0 0 1 0 0 0 0 0 1 0 0 0 1 0 0
1 1 0 0 0 0 0 0 1 0 0 0 0 0
[407] 0 0 1 0 0 0 0 0 1 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 1 0 0 0 0 0 1 0 0 1 0 1 0 0 1
0 1 0 0 0 0 0 0 0 0 0 1 1 0 0
[465] 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 1 0 1 0 0 1 0 0 0 0 0 1 1 0 1 0 1 0 0 0 0
0 1 0 0 1 0 1 0 1 1 0 0 0 1
[523] 0 0 0 0 0 0 0 0 0 0 0 1 0 1 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1
1 1 0
```

Vemos cómo se distriben las etiquetas y si los datos están balanceados.

Hide

```
table(diagnosis) # cuantos observaciones tienen diagnostico maligno
```

```
diagnosis
  0   1
357 212
```

Evaluación utilizando el conocimiento experto

Resultados del kMeans para k=2 respecto a la etiqueta de los datos. ¿Qué observamos?

Hide

```
table(km_valor_2_wisc_data_reduccion$cluster, diagnosis)
```

```
   diagnosis
      0   1
  1  14 175
  2 343  37
```

Resultados del kMeans para k=4 respecto a la etiqueta de los datos. ¿Qué observamos?

Hide

```
table(km_valor_4_wisc_data_reduccion$cluster, diagnosis)
```

```
  diagnosis
     0   1
1    0  37
2   36  54
3    0  97
4  321  24
```

Resultados del jerárquico respecto a la etiqueta de los datos. ¿Qué observamos?

Hide

```
table(wisc_hclust_clusters, diagnosis)
```

```
                   diagnosis
wisc_hclust_clusters   0   1
                   1  12 165
                   2   2   5
                   3 343  40
                   4   0   2
```

Se puede observar que algunos puntos no se separan del todo bien teniendo en cuenta la información de las etiquetas y las de los clústeres. ¿Qué ocurre en esos puntos? Quizás sean puntos interesantes para un estudio con más detalles.

Gráficas del PCA con el color de las etiquetas.

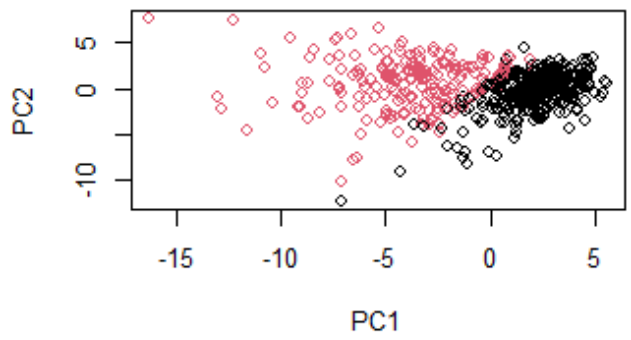
Por último, como curiosidad, Repetimos las gráficas que hicimos antes de los ejes coordenados que se obtenían con el PCA, pero coloreando según la etiqueta.

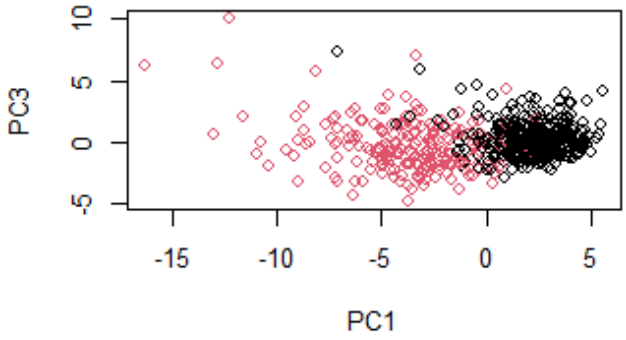
Hide

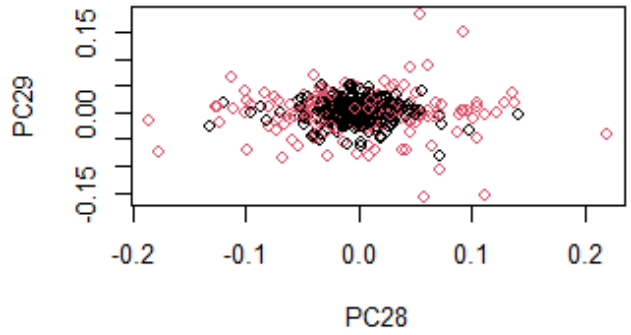
```
par(mfrow = c(2,2))
plot(wisc_pca$x[,c(1,2)],col=(diagnosis+1), xlab = "PC1",ylab = "PC2")
plot(wisc_pca$x[,c(1,3)],col=(diagnosis+1), xlab = "PC1",ylab = "PC3")
```

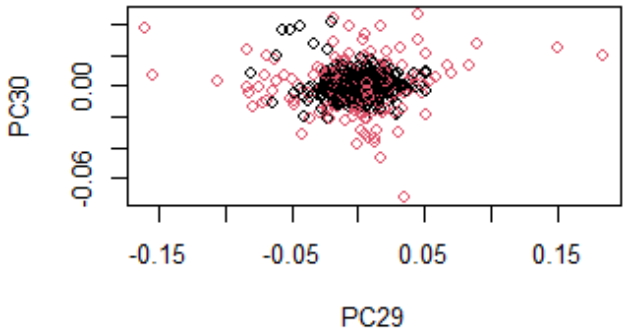
Hide

```
plot(wisc_pca$x[,c(28,29)],col=(diagnosis+1), xlab = "PC28",ylab = "PC29")
plot(wisc_pca$x[,c(29,30)],col=(diagnosis+1), xlab = "PC29",ylab = "PC30")
```









Observando cómo se distribuyen las etiquetas en los ejes coordenados del PCA podemos intuir cuáles son los puntos que hemos comentado en el apartado anterior. Puntos que, teniendo distinta etiqueta, tienen – desde un punto de vista de los datos – una situación bastante parecida.

Podemos elucubrar que … ¿Y si las etiquetas no son totalmente disjuntas? ¿Y si hay pacientes enfermos, pero poco enfermos, y pacientes sanos, pero a punto de estar enfermos?