Parcial2Computacion

Miguel Diaz

2022-06-09

Asignación

- Correr el mismo código de clase, para Chroma y Hue
- Caracterizar los Cluster
- Utilizar algún gráfico que muestre el número de cluster
- Pasar de coordenadas Lab a RGB (Investigar) y hacer los cluster con las coordanadas RGB
- Existe alguna relación del Chroma y Hue para coordenadas RGB, si es cierto, entonces realiza los cluster con RGB
- Pueden colocar otro tipo de coordenadas (ej. HLS, HVS)

```
dfcie <- read_excel("C:\\Users\\usuario\\Desktop\\RCOMPUTACION\\Clases r\\Cielab_tueste_cafe+cyh.xlsx")
```

```
C <- sqrt(dfcie$a^2+dfcie$b^2)
h <- atan(dfcie$b/dfcie$a)
Cyh <- matrix(c(C,h),
              ncol=2)
Cyh
```

```
##            [,1]      [,2]
##   [1,] 32.29018 0.7480722
##   [2,] 29.55308 0.7911271
##   [3,] 33.36546 0.7977737
##   [4,] 29.34204 0.7693511
##   [5,] 28.45274 0.8144945
##   [6,] 30.23399 0.7605873
##   [7,] 34.35739 0.7826524
##   [8,] 32.15757 0.8387400
##   [9,] 27.88013 0.7701170
##  [10,] 32.06586 0.7674516
##  [11,] 33.46825 0.7817533
##  [12,] 32.67243 0.7917086
##  [13,] 32.74687 0.8010740
##  [14,] 30.09405 0.8133458
##  [15,] 31.34843 0.7700990
##  [16,] 32.92919 0.7825687
##  [17,] 31.13403 0.7701947
##  [18,] 29.54272 0.7817248
```

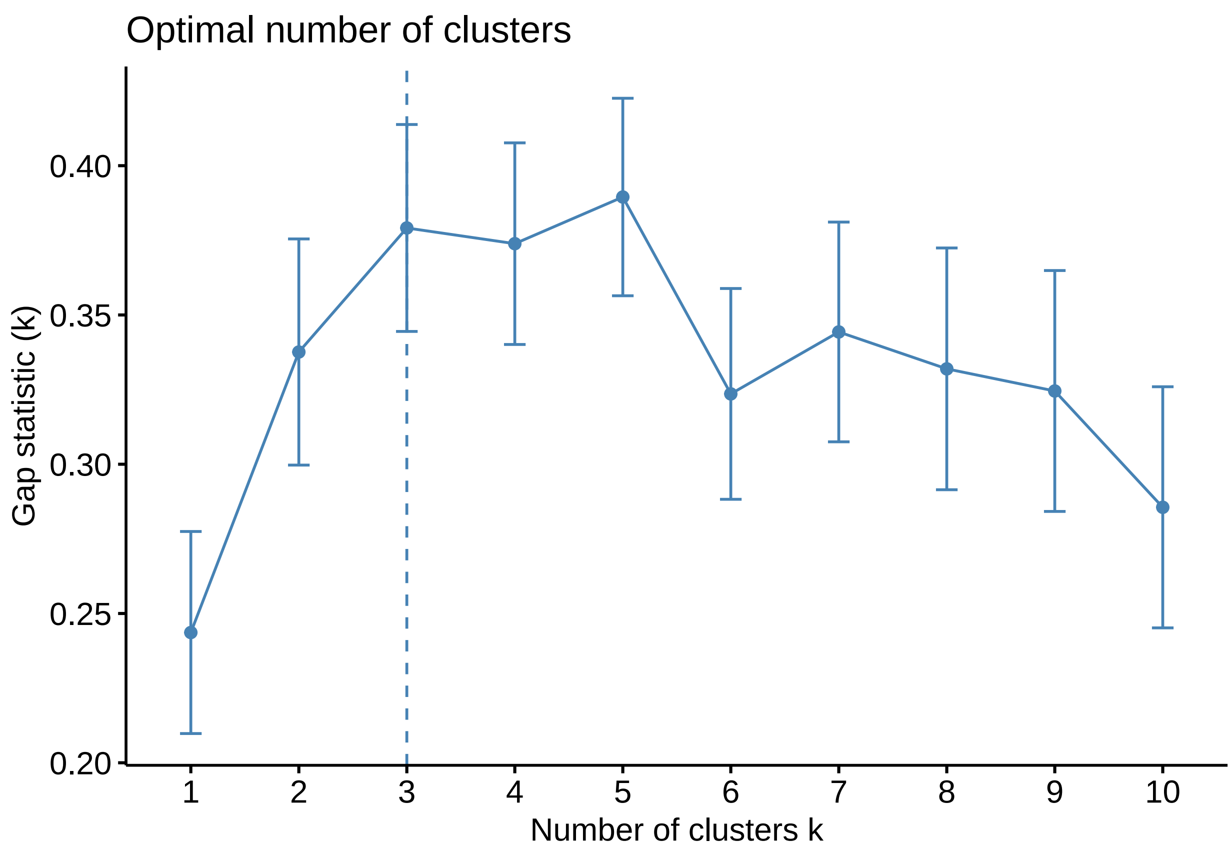
```
## [19,] 30.80794 0.7881335
## [20,] 32.24589 0.8290143
## [21,] 30.96257 0.8165097
## [22,] 30.15301 0.8367726
## [23,] 33.32562 0.7810805
## [24,] 26.48464 0.7562150
## [25,] 32.83202 0.8073524
## [26,] 28.20362 0.7470730
## [27,] 32.66340 0.7858145
## [28,] 27.32314 0.7998872
## [29,] 25.28551 0.7884349
## [30,] 31.66948 0.7643224
## [31,] 36.54232 0.8721633
## [32,] 38.18454 0.8112716
## [33,] 37.89272 0.8630906
## [34,] 42.59641 0.8291944
## [35,] 36.61186 0.8418260
## [36,] 40.35032 0.8293775
## [37,] 40.26108 0.8146810
## [38,] 40.36756 0.8578553
## [39,] 39.28339 0.8457257
## [40,] 40.68284 0.8432812
## [41,] 36.28885 0.8548275
## [42,] 41.91576 0.8495827
## [43,] 39.12490 0.8400389
## [44,] 38.46102 0.8481779
## [45,] 34.47377 0.8558675
## [46,] 38.13114 0.8467254
## [47,] 38.04637 0.8507228
## [48,] 38.89867 0.8298000
## [49,] 35.29301 0.8652878
## [50,] 39.56621 0.8334756
## [51,] 38.53694 0.8181597
## [52,] 38.18086 0.8268372
## [53,] 38.12356 0.8264071
## [54,] 35.88550 0.8406499
## [55,] 36.56295 0.8625368
## [56,] 38.29227 0.8415774
## [57,] 37.09680 0.8418444
## [58,] 36.98528 0.8234115
## [59,] 37.20435 0.8413033
## [60,] 41.26583 0.8407752
## [61,] 19.87163 0.7475825
## [62,] 23.49304 0.8482371
## [63,] 24.55295 0.8188454
## [64,] 22.50252 0.8719029
## [65,] 22.07605 0.7553501
## [66,] 21.61435 0.7373361
## [67,] 23.05659 0.7620604
## [68,] 24.97770 0.8216690
## [69,] 22.47735 0.8242375
## [70,] 29.42129 0.7688550
## [71,] 20.11665 0.8209366
## [72,] 21.99449 0.7614353
```

```
##  [73,] 20.40406 0.7632498
##  [74,] 23.83570 0.7976805
##  [75,] 27.48433 0.7611407
##  [76,] 25.75828 0.8095348
##  [77,] 25.94727 0.8074646
##  [78,] 22.47398 0.8348585
##  [79,] 20.21408 0.8780080
##  [80,] 23.66820 0.7882196
##  [81,] 21.75814 0.7925799
##  [82,] 21.52435 0.7554670
##  [83,] 22.30371 0.7391472
##  [84,] 24.88019 0.8051933
##  [85,] 25.14254 0.7342786
##  [86,] 23.57896 0.7749939
##  [87,] 20.44878 0.8279841
##  [88,] 21.69991 0.7180264
##  [89,] 25.02320 0.8236181
##  [90,] 20.74814 0.8309550
##  [91,] 28.40329 0.7936993
##  [92,] 24.57498 0.7857908
##  [93,] 22.44342 0.7691486
##  [94,] 23.09517 0.8026353
##  [95,] 28.46398 0.7757886
##  [96,] 22.60838 0.7497393
##  [97,] 24.69810 0.8400200
##  [98,] 24.81095 0.7677115
##  [99,] 23.62731 0.8015076
## [100,] 24.08223 0.7869244
## [101,] 27.54536 0.7564268
## [102,] 24.93081 0.7557990
## [103,] 25.98874 0.7667001
## [104,] 25.81900 0.7055609
## [105,] 27.15064 0.7600131
## [106,] 24.98160 0.7949536
## [107,] 27.08809 0.8365323
## [108,] 26.48606 0.8332368
## [109,] 25.34591 0.8311150
## [110,] 24.96307 0.8038209
## [111,] 23.48658 0.8058029
## [112,] 21.63534 0.7280711
## [113,] 28.83561 0.8085676
## [114,] 23.18006 0.8365677
## [115,] 25.78536 0.8391969
## [116,] 24.80234 0.8125544
## [117,] 26.73317 0.7841935
## [118,] 23.49753 0.7890172
## [119,] 28.58817 0.7814427
## [120,] 22.22414 0.8237892
```

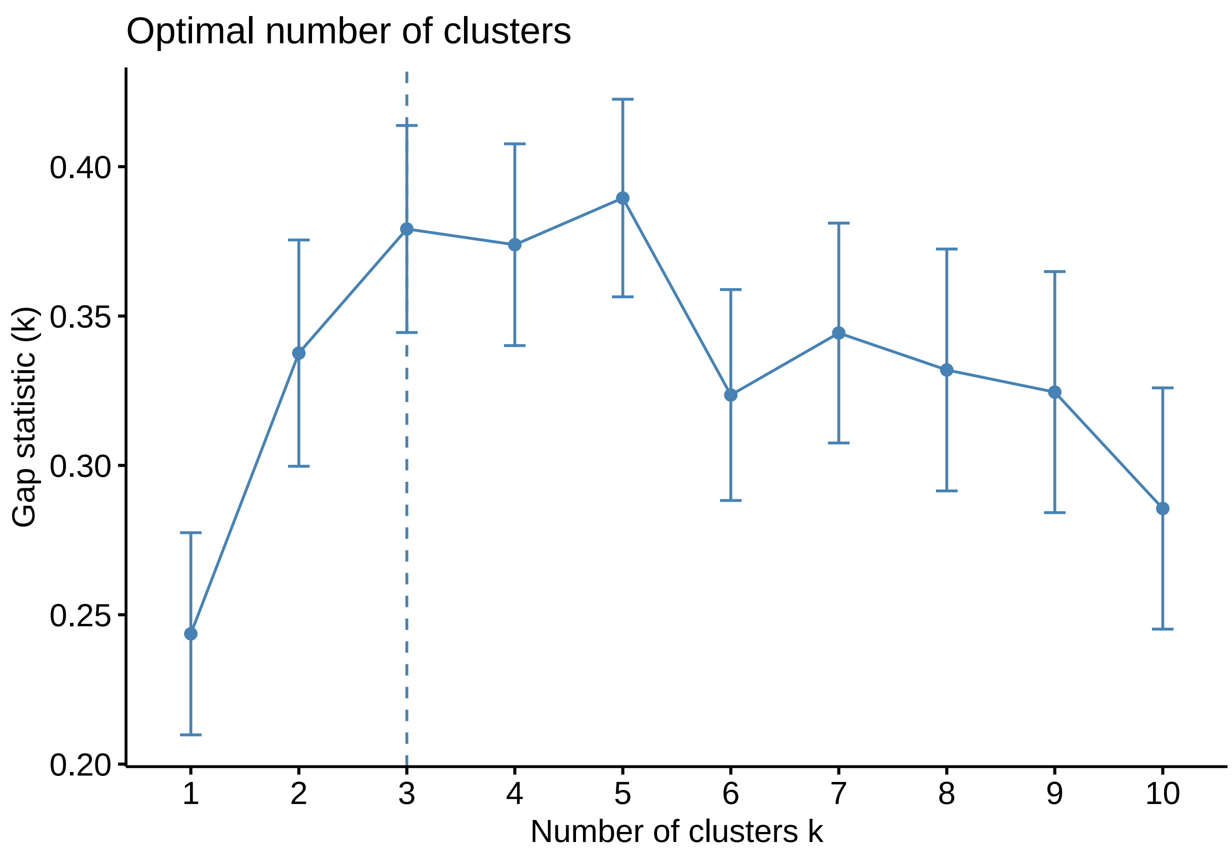
```r
#Haciendo una matriz Chroma y hue en R, sin embargo, modifique el excel para tenerlos en el doc, por un
```

```r
## Número óptimo de clusters
M = Cyh
Ms = scale(M)
```

```
fviz_nbclust(Ms,
             FUNcluster = kmeans,
             method = 'gap_stat',
             diss = get_dist(Ms,
                             'euclidean'))
```



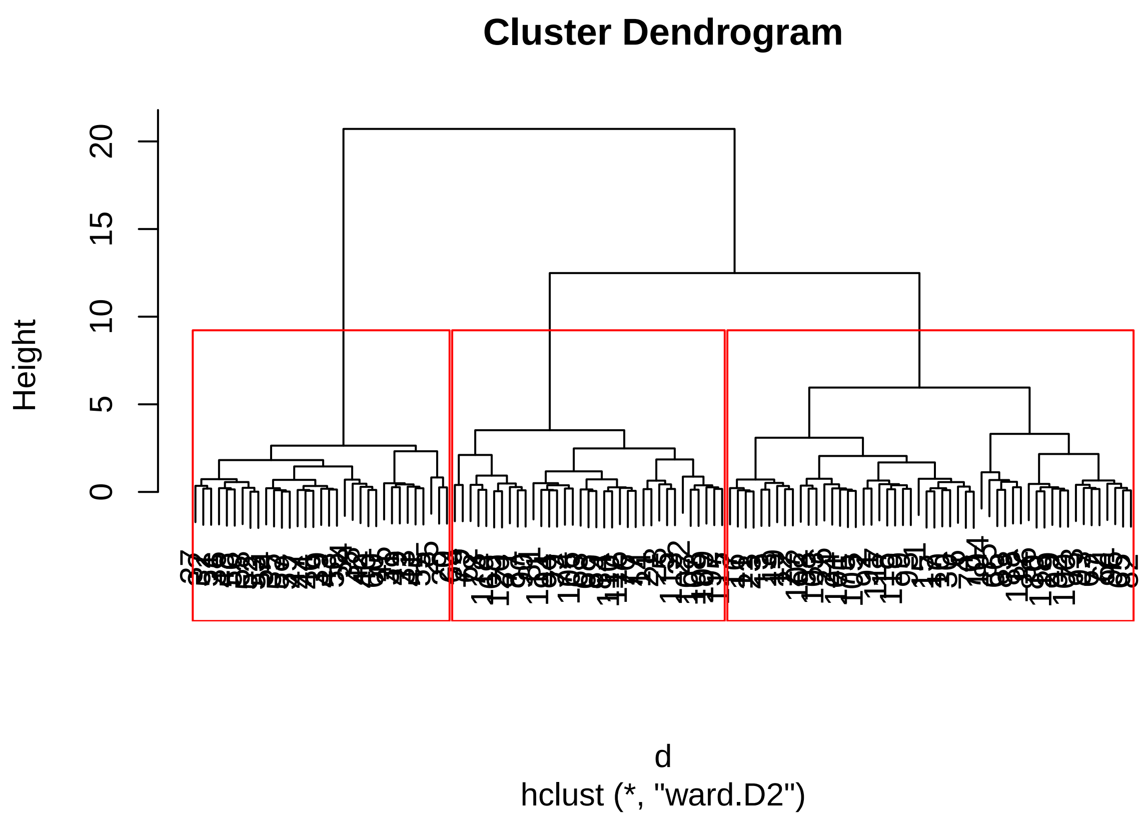

```
## Número óptimo de clusters
M = dfcie[ ,-1]
M2 = M [ ,-1]
M3 = M2 [ ,-1]
M4 = M3 [ ,-1]
Ms = scale(M4)
fviz_nbclust(Ms,
             FUNcluster = kmeans,
             method = 'gap_stat',
             diss = get_dist(Ms,
                             'euclidean'))
```




```
clus1 = kmeans(Ms, 3)
M4$cluster <- clus1$cluster
M4 |>
  group_by(cluster) |>
  summarise(media_C = mean(C),
            media_h = mean(h),
            desv_C = sd(C),
            desv_h = sd(h),
            coeV_C = 100 * desv_C/media_C,
            coeV_h = 100 *desv_h/media_h,)
```

```
## # A tibble: 3 x 7
##   cluster media_C media_h desv_C desv_h coeV_C coeV_h
##     <int>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1       1    37.8   0.840   2.54 0.0158   6.71   1.88
## 2       2    24.7   0.822   2.79 0.0197  11.3    2.40
## 3       3    27.2   0.767   4.16 0.0205  15.3    2.67
```

```
df_scale <- scale(M4)
d <- dist(df_scale, method = "euclidean")
fit <- hclust(d, method="ward.D2")
plot(fit)
groups <- cutree(fit, k=3)
rect.hclust(fit, k=3, border="red")
```




```r
tapply(M4$C, groups, mean)
```

```
##        1        2        3
## 27.16321 24.70837 37.82856
```

```r
tapply(M4$h, groups, mean)
```

```
##         1         2         3
## 0.7673173 0.8218576 0.8400479
```

```r
#PASAR A RGB
dfcie1 <- dfcie [ ,-4]
dfcie2 <- dfcie1 [ ,-4]
dfcie3 <- dfcie2 [ ,-4]
dfcie3
```

```
## # A tibble: 120 x 3
##         L     a     b
##     <dbl> <dbl> <dbl>
##  1  15.2  23.7  22.0
##  2  12.3  20.8  21.0
##  3  13.5  23.3  23.9
##  4  13.6  21.1  20.4
##  5  13.3  19.5  20.7
```

```
## 6   12.4  21.9  20.8
## 7   16.4  24.4  24.2
## 8   14.1  21.5  23.9
## 9   11.8  20.0  19.4
## 10  14.6  23.1  22.3
## # ... with 110 more rows
```

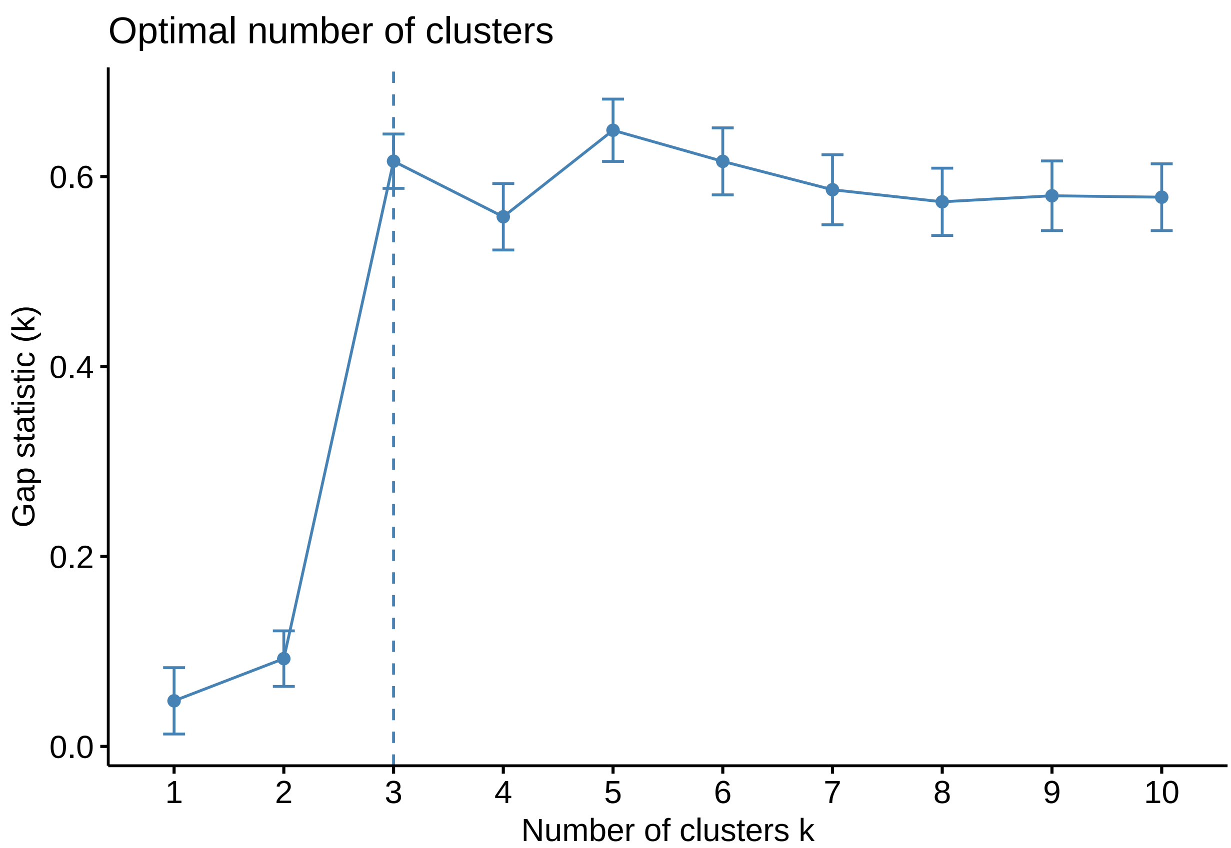
```r
#PASAR A RGB
library(farver)
dfrgb<- convert_colour(dfcie3,'lab','rgb')
dfrgb
```

```
##              r        g           b
## 1    72.67764 20.92596  2.73267573
## 2    62.80198 17.51391  0.00000000
## 3    68.68871 17.37661  0.00000000
## 4    65.91809 20.08825  0.79302516
## 5    63.83378 20.99274  0.00000000
## 6    63.92127 16.40077  0.00000000
## 7    76.81804 22.85914  1.39417457
## 8    68.36776 20.70484  0.00000000
## 9    60.39985 17.12378  0.00000000
## 10   70.80204 20.28769  0.07709687
## 11   68.41939 16.22298  0.00000000
## 12   71.29125 20.75246  0.00000000
## 13   70.98326 20.73070  0.00000000
## 14   67.93681 22.37354  0.28745091
## 15   70.76505 21.49941  1.96940393
## 16   69.03642 17.78844  0.00000000
## 17   65.80367 17.11087  0.00000000
## 18   63.92970 18.29821  0.00000000
## 19   67.37263 19.83552  0.00000000
## 20   68.77106 20.55613  0.00000000
## 21   68.77614 21.98758  0.00000000
## 22   68.74562 23.88500  0.66161441
## 23   70.06856 18.07190  0.00000000
## 24   60.24529 18.76858  1.01975592
## 25   69.90397 19.83317  0.00000000
## 26   63.94644 19.29920  1.85133011
## 27   73.95803 23.00820  2.15978757
## 28   56.36678 14.91563  0.00000000
## 29   54.18067 15.58712  0.00000000
## 30   66.11327 16.28020  0.00000000
## 31  106.87410 49.58532 23.16851804
## 32   99.66334 38.44458 16.02761489
## 33  108.39261 48.66785 21.92270089
## 34  121.20785 51.23165 23.59925178
## 35  103.85262 45.55531 21.39241939
## 36  118.66186 52.25492 25.98962496
## 37  113.77134 47.52208 22.82939697
## 38  116.74964 52.11466 23.68627677
## 39  117.86707 53.88962 27.00060190
## 40  117.97379 51.94398 24.41957047
## 41  103.78617 46.52540 21.64925140
```

```
## 42  123.55288 55.26656 26.13137881
## 43  114.39631 50.90986 24.77053469
## 44  112.80462 50.87835 24.61266124
## 45  106.72050 51.42888 27.39301640
## 46  111.32003 50.00690 24.13744731
## 47  110.29886 49.45873 23.40996336
## 48  111.10117 47.94954 22.93801277
## 49  108.40412 52.18133 26.93807366
## 50  114.32981 49.91905 24.04580901
## 51  113.66298 50.03531 25.91257888
## 52  111.41798 49.06226 24.62848560
## 53  109.30148 47.34455 23.11818144
## 54  104.37160 46.91030 23.20142866
## 55  112.99048 54.24933 28.10964800
## 56  111.01404 49.28273 23.72082811
## 57  108.41972 48.73494 24.00020677
## 58  113.44339 52.26206 28.54764446
## 59  111.69406 51.31429 26.34352683
## 60  118.69793 51.61667 23.91064680
## 61   59.19814 27.32729 17.59884964
## 62   65.80884 30.53196 15.13057211
## 63   68.59459 30.80435 16.05367995
## 64   66.35272 32.86064 17.48510704
## 65   64.18658 28.78635 17.87586420
## 66   65.27337 29.98946 19.95420083
## 67   67.47122 30.42804 18.85395779
## 68   71.76801 33.06048 18.05216165
## 69   67.13516 32.38932 18.78026538
## 70   83.34513 35.30951 20.25202425
## 71   64.33502 32.94329 20.79545176
## 72   68.73709 32.95713 21.74966318
## 73   60.78390 28.28824 17.84776104
## 74   69.75101 32.20084 18.82777130
## 75   77.26769 32.63654 19.03741349
## 76   66.14862 26.77646 10.50587565
## 77   71.45927 31.11504 16.08009105
## 78   67.31535 32.80723 18.81152109
## 79   59.75839 30.05384 15.87061671
## 80   68.95957 31.50741 18.60522686
## 81   64.59914 30.40266 18.33234091
## 82   60.21391 26.06094 14.99721535
## 83   68.71319 32.02094 21.51584748
## 84   70.52051 31.66818 17.40027451
## 85   77.51342 35.42468 23.71882259
## 86   69.28028 31.57800 19.23674232
## 87   60.49671 29.31402 16.64622363
## 88   66.77907 30.76577 21.30820563
## 89   74.96630 35.78173 20.51951416
## 90   62.07350 30.37804 17.52859297
## 91   77.83784 32.84928 17.20769899
## 92   70.97274 31.95185 18.67594824
## 93   64.70535 29.03691 17.42533255
## 94   66.51603 30.51775 17.31870484
## 95   77.72817 32.10519 17.35750013
```

```
## 96   68.79068 31.89727 20.90211736
## 97   74.81262 36.52343 20.76098892
## 98   75.73127 35.22184 22.32960723
## 99   73.01286 35.36284 21.75627793
## 100  73.68432 34.95950 21.69207126
## 101  74.36798 29.93017 16.55624451
## 102  73.52240 32.86448 20.57650578
## 103  78.49626 35.92212 22.47465071
## 104  76.72987 33.04649 22.48265074
## 105  79.60777 35.05733 21.44949264
## 106  76.59288 36.44203 22.29225342
## 107  78.48510 36.48283 19.39674118
## 108  71.26705 30.98873 14.26218059
## 109  72.64089 33.59417 17.95779029
## 110  78.00722 37.91390 23.31575167
## 111  65.00828 28.75483 14.93933244
## 112  71.53906 35.16540 25.09611230
## 113  87.20987 40.68375 23.53446426
## 114  66.81742 31.51330 16.95757930
## 115  72.59407 33.22025 16.93933807
## 116  70.11708 31.62267 17.07399406
## 117  74.04029 31.59935 17.28935300
## 118  71.08770 33.57784 20.62737112
## 119  83.02760 36.60495 21.21118220
## 120  70.31887 35.43990 21.82361742
```

```r
## Cluster para RGB
Msrgb = scale(dfrgb)
fviz_nbclust(Msrgb,
             FUNcluster = kmeans,
             method = 'gap_stat',
             diss = get_dist(Msrgb,
                             'euclidean'))
```

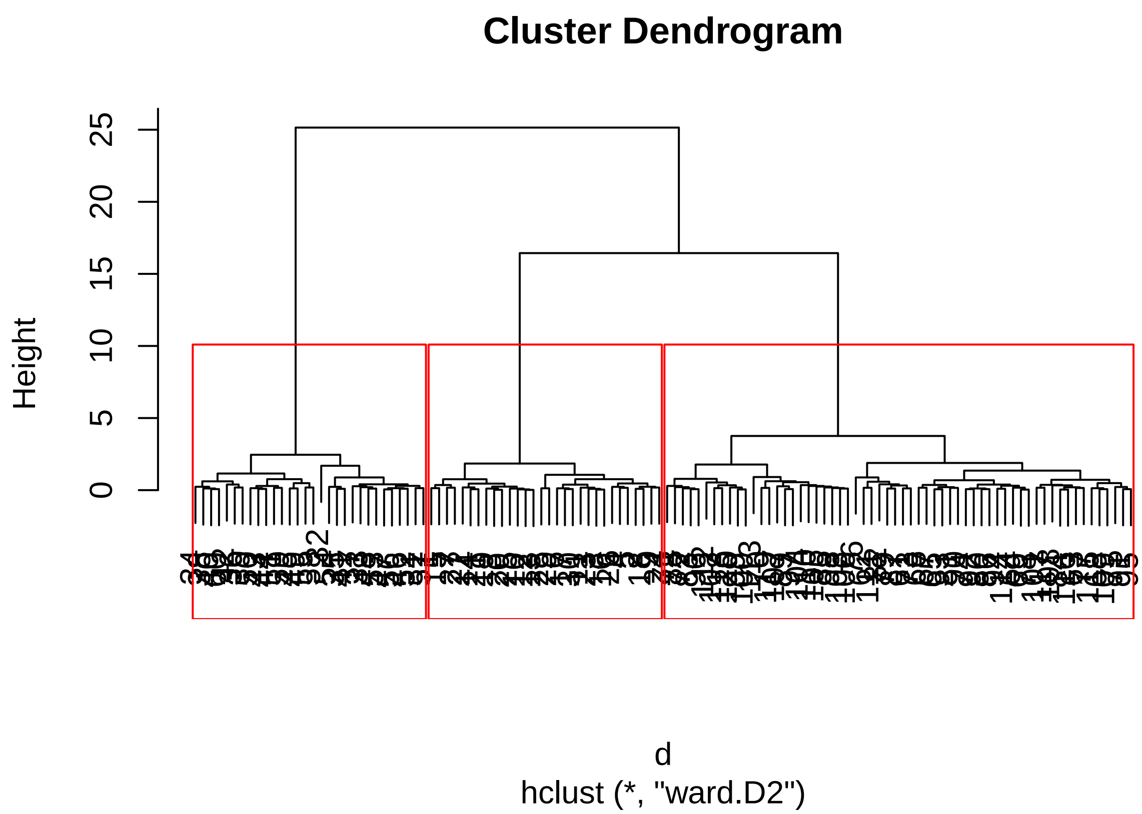



```r
#Hay 3 cluster para RGB tambien
clus1 = kmeans(Msrgb, 3)
dfrgb$cluster <- clus1$cluster
dfrgb |>
  group_by(cluster) |>
  summarise(media_r = mean(r),
            media_g = mean(g),
            media_b = mean(b),
            desv_r = sd(r),
            desv_g = sd(g),
            desv_b = sd(b),
            coeV_r = 100 * desv_r/media_r,
            coeV_g = 100 *desv_g/media_g,
            coeV_b = 100 *desv_b/media_b)
```

```
## # A tibble: 3 x 10
##   cluster media_r media_g media_b desv_r desv_g desv_b coeV_r coeV_g coeV_b
##     <int>   <dbl>   <dbl>   <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>  <dbl>
## 1       1    70.7    32.5  19.0     6.16   2.83  2.73    8.72   8.72  14.3
## 2       2    67.0    19.4   0.432   4.93   2.39  0.788   7.36  12.3  183.
## 3       3   112.     49.9  24.3     5.48   3.17  2.42    4.90   6.35   9.97
```

```r
df_scale1 <- scale(dfrgb)
d <- dist(df_scale1, method = "euclidean")
fit <- hclust(d, method="ward.D2")
```

```r
plot(fit)
groups <- cutree(fit, k=3)
rect.hclust(fit, k=3, border="red")
```




```r
tapply(dfrgb$r, groups, mean)
```

```
##         1         2         3
##  67.02811 111.89141  70.73459
```

```r
tapply(dfrgb$g, groups, mean)
```

```
##        1        2        3
## 19.36928 49.88490 32.45031
```

```r
tapply(dfrgb$b, groups, mean)
```

```
##          1          2          3
##  0.4315438 24.2518433 19.0175785
```