Mineria de Datos

Luis Enrique Olascoaga Dominguez

22/11/2022

PEC2

Obtenemos el set de datos llamado “Hawks”

Obtenemos librerias a utilizar

```
if (!require('cluster')) install.packages('cluster')
```

```
## Loading required package: cluster
```

```
library(cluster)

if (!require('Stat2Data')) install.packages('Stat2Data')
```

```
## Loading required package: Stat2Data
```

```
library(Stat2Data)

if (!require('ggplot2')) install.packages('ggplot2')
```

```
## Loading required package: ggplot2
```

```
library(ggplot2)

if (!require('fpc')) install.packages('fpc')
```

```
## Loading required package: fpc
```

```
library(fpc)

if (!require('dbscan')) install.packages('dbscan')
```

```
## Loading required package: dbscan
```

```
##
## Attaching package: 'dbscan'
```

```
## The following object is masked from 'package:fpc':
##
##     dbscan

## The following object is masked from 'package:stats':
##
##     as.dendrogram
```

```
library(dbscan)

if (!require('tidyverse')) install.packages('tidyverse')
```

```
## Loading required package: tidyverse

## -- Attaching packages --------------------------------------- tidyverse 1.3.2 --
## v tibble  3.1.8      v dplyr   1.0.10
## v tidyr   1.2.1      v stringr 1.4.1
## v readr   2.1.3      v forcats 0.5.2
## v purrr   0.3.5
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
library(tidyverse)


if (!require('factoextra')) install.packages('factoextra')
```

```
## Loading required package: factoextra
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
library(factoextra)

#Obtenemos el set de datos llamado "Hawks"
data("Hawks")
summary(Hawks)
```

```
##      Month             Day             Year       CaptureTime   ReleaseTime
##  Min.   : 8.000   Min.   : 1.00   Min.   :1992   11:35  : 14          :842
##  1st Qu.: 9.000   1st Qu.: 9.00   1st Qu.:1995   13:30  : 14   11:00  :  2
##  Median :10.000   Median :16.00   Median :1999   11:45  : 13   11:35  :  2
##  Mean   : 9.843   Mean   :15.74   Mean   :1998   12:10  : 13   12:05  :  2
##  3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:2001   14:00  : 13   12:50  :  2
##  Max.   :11.000   Max.   :31.00   Max.   :2003   13:05  : 12   13:32  :  2
##                                                  (Other):829   (Other): 56
##       BandNumber  Species  Age     Sex          Wing            Weight
##            :  2   CH: 70   A:224    :576   Min.   : 37.2   Min.   :  56.0
##  1142-09240:  1   RT:577   I:684   F:174   1st Qu.:202.0   1st Qu.: 185.0
##  1142-09241:  1   SS:261           M:158   Median :370.0   Median : 970.0
##  1142-09242:  1                           Mean   :315.6   Mean   : 772.1
##  1142-18229:  1                           3rd Qu.:390.0   3rd Qu.:1120.0
```

```
##  1142-19209:  1                          Max.   :480.0   Max.   :2030.0
##  (Other)   :901                          NA's   :1       NA's   :10
##      Culmen         Hallux            Tail        StandardTail
##  Min.   : 8.6   Min.   :  9.50   Min.   :119.0   Min.   :115.0
##  1st Qu.:12.8   1st Qu.: 15.10   1st Qu.:160.0   1st Qu.:162.0
##  Median :25.5   Median : 29.40   Median :214.0   Median :215.0
##  Mean   :21.8   Mean   : 26.41   Mean   :198.8   Mean   :199.2
##  3rd Qu.:27.3   3rd Qu.: 31.40   3rd Qu.:225.0   3rd Qu.:226.0
##  Max.   :39.2   Max.   :341.40   Max.   :288.0   Max.   :335.0
##  NA's   :7      NA's   :6                        NA's   :337
##      Tarsus        WingPitFat        KeelFat           Crop
##  Min.   :24.70   Min.   :0.0000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:55.60   1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:0.0000
##  Median :79.30   Median :1.0000   Median :2.000   Median :0.0000
##  Mean   :71.95   Mean   :0.7922   Mean   :2.184   Mean   :0.2345
##  3rd Qu.:87.00   3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:0.2500
##  Max.   :94.00   Max.   :3.0000   Max.   :4.000   Max.   :5.0000
##  NA's   :833     NA's   :831      NA's   :341     NA's   :343
```

```
#{r pressure, echo=FALSE}
```

EDA (exploratory data analysis)

Obtenemos nombre de las columnas y su tipo de dato

```
str(Hawks)
```

```
## 'data.frame':    908 obs. of  19 variables:
##  $ Month       : int  9 9 9 9 9 9 9 9 9 9 ...
##  $ Day         : int  19 22 23 23 27 28 28 29 29 30 ...
##  $ Year        : int  1992 1992 1992 1992 1992 1992 1992 1992 1992 1992 ...
##  $ CaptureTime : Factor w/ 308 levels " ","1:15","1:31",..: 181 25 138 42 62 71 181 88 261 192 ...
##  $ ReleaseTime : Factor w/ 60 levels "","  ","10:20",..: 1 2 2 2 2 2 2 2 2 2 ...
##  $ BandNumber  : Factor w/ 907 levels " ","1142-09240",..: 856 857 858 809 437 280 859 860 861 281 .
##  $ Species     : Factor w/ 3 levels "CH","RT","SS": 2 2 2 1 3 2 2 2 2 2 ...
##  $ Age         : Factor w/ 2 levels "A","I": 2 2 2 2 2 2 2 1 1 2 ...
##  $ Sex         : Factor w/ 3 levels "","F","M": 1 1 1 2 2 1 1 1 1 1 ...
##  $ Wing        : num  385 376 381 265 205 412 370 375 412 405 ...
##  $ Weight      : int  920 930 990 470 170 1090 960 855 1210 1120 ...
##  $ Culmen      : num  25.7 NA 26.7 18.7 12.5 28.5 25.3 27.2 29.3 26 ...
##  $ Hallux      : num  30.1 NA 31.3 23.5 14.3 32.2 30.1 30 31.3 30.2 ...
##  $ Tail        : int  219 221 235 220 157 230 212 243 210 238 ...
##  $ StandardTail: int  NA NA NA NA NA NA NA NA NA NA ...
##  $ Tarsus      : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ WingPitFat  : int  NA NA NA NA NA NA NA NA NA NA ...
##  $ KeelFat     : num  NA NA NA NA NA NA NA NA NA NA ...
##  $ Crop        : num  NA NA NA NA NA NA NA NA NA NA ...
```

Observamos en el preview de los datos que las columnas StandardTail, Tarsus, WingPitFat, KeelFat y Crop, contienen datos NA, vamos a revisarlos y a removerlos ya que no se ocuparan para este análsis. Las columnas numericas a utilizar serán:

Wing: Longitud (en mm) de la pluma principal del ala desde la punta hasta la muñeca a la que se une
Weight: Peso corporal (en gm)
Culmen: Longitud (en mm) del pico superior desde la punta hasta donde choca con la parte carnosa del ave
Hallux: Longitud (en mm) de la garra asesina

Y la calumna que se utilizará para comparar nuestros clusters posteriormente será: Species: CH=Halcón de Cooper, RT=Colirrojo, SS=Gavilán

La descripción del layout fue obtenida desde: https://vincentarelbundock.github.io/Rdatasets/doc/Stat2Data/Hawks.html

Para nuestro análsis vamos a generar 1 dataframe llamado: hawks_k_means: Tendra las columnas Wing, Weigh, Culmen y Hallux

Generamos el dataframe k-means

```
hawks_k_means <- na.omit(Hawks[,10:13])
summary(hawks_k_means)
```

```
##       Wing           Weight           Culmen          Hallux
##  Min.   : 37.2   Min.   :  56.0   Min.   : 8.60   Min.   :  9.50
##  1st Qu.:202.0   1st Qu.: 185.0   1st Qu.:12.80   1st Qu.: 15.10
##  Median :370.0   Median : 970.0   Median :25.50   Median : 29.40
##  Mean   :315.9   Mean   : 771.6   Mean   :21.81   Mean   : 26.41
##  3rd Qu.:390.0   3rd Qu.:1120.0   3rd Qu.:27.35   3rd Qu.: 31.40
##  Max.   :480.0   Max.   :2030.0   Max.   :39.20   Max.   :341.40
```

```
hawks_k_original <- na.omit(Hawks[,7:13])

hawks_k_means_dbscan_wing_weight <- na.omit(Hawks[,10:11])
hawks_k_means_dbscan_hallux_weight <- na.omit(Hawks[,11:13])
hawks_k_means_dbscan_hallux_weight <- na.omit(hawks_k_means_dbscan_hallux_weight[,-2])
```

Cuando reducimos la dimensionalidad de nuestro dataframe, solo dejamos las 4 columnas a ocupar para el modelo k-means. En el summary encontramos que no hay valores en NA o nulos y todos los valores son númericos.

Revisamos nuevamente el set de datos original

```
summary(Hawks)
```

```
##      Month             Day             Year       CaptureTime   ReleaseTime
##  Min.   : 8.000   Min.   : 1.00   Min.   :1992   11:35  : 14          :842
##  1st Qu.: 9.000   1st Qu.: 9.00   1st Qu.:1995   13:30  : 14   11:00  :  2
##  Median :10.000   Median :16.00   Median :1999   11:45  : 13   11:35  :  2
##  Mean   : 9.843   Mean   :15.74   Mean   :1998   12:10  : 13   12:05  :  2
##  3rd Qu.:10.000   3rd Qu.:23.00   3rd Qu.:2001   14:00  : 13   12:50  :  2
##  Max.   :11.000   Max.   :31.00   Max.   :2003   13:05  : 12   13:32  :  2
##                                                  (Other):829   (Other): 56
##       BandNumber  Species  Age     Sex          Wing            Weight
##            :  2   CH: 70   A:224    :576   Min.   : 37.2   Min.   :  56.0
##  1142-09240:  1   RT:577   I:684   F:174   1st Qu.:202.0   1st Qu.: 185.0
##  1142-09241:  1   SS:261           M:158   Median :370.0   Median : 970.0
##  1142-09242:  1                           Mean   :315.6   Mean   : 772.1
##  1142-18229:  1                           3rd Qu.:390.0   3rd Qu.:1120.0
```

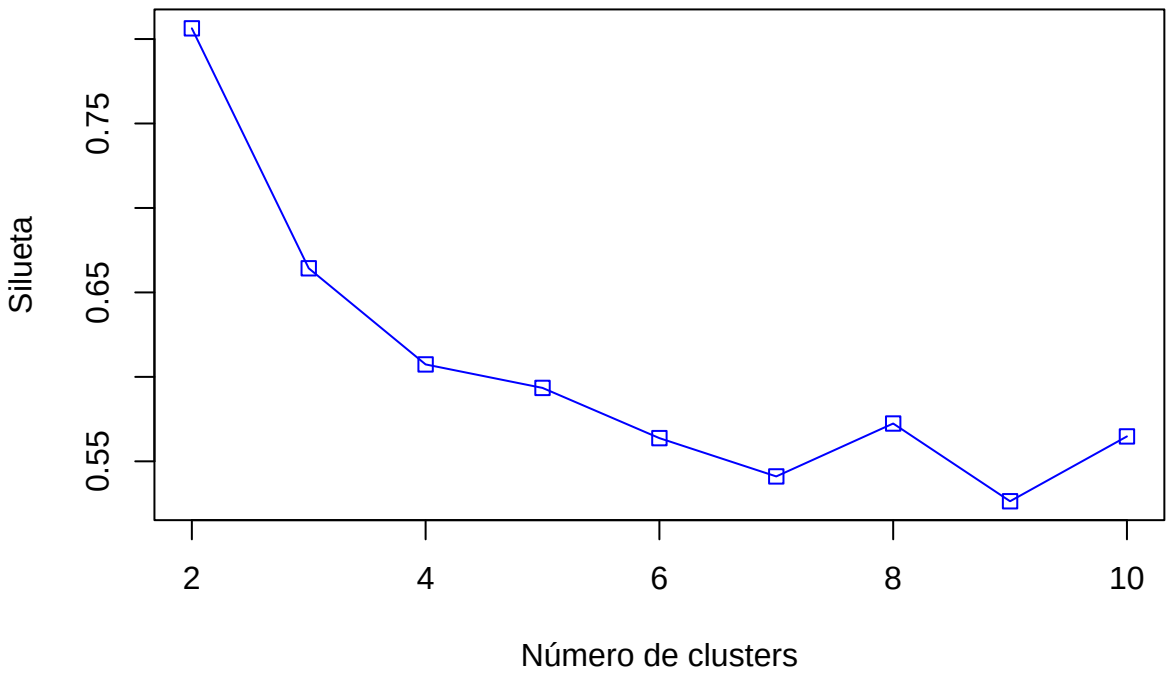
```
##  1142-19209:  1                                Max.   :480.0   Max.   :2030.0
##  (Other)   :901                                NA's   :1       NA's   :10
##      Culmen          Hallux             Tail         StandardTail
##  Min.   : 8.6   Min.   :  9.50   Min.   :119.0   Min.   :115.0
##  1st Qu.:12.8   1st Qu.: 15.10   1st Qu.:160.0   1st Qu.:162.0
##  Median :25.5   Median : 29.40   Median :214.0   Median :215.0
##  Mean   :21.8   Mean   : 26.41   Mean   :198.8   Mean   :199.2
##  3rd Qu.:27.3   3rd Qu.: 31.40   3rd Qu.:225.0   3rd Qu.:226.0
##  Max.   :39.2   Max.   :341.40   Max.   :288.0   Max.   :335.0
##  NA's   :7      NA's   :6                        NA's   :337
##      Tarsus        WingPitFat        KeelFat           Crop
##  Min.   :24.70   Min.   :0.0000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:55.60   1st Qu.:0.0000   1st Qu.:2.000   1st Qu.:0.0000
##  Median :79.30   Median :1.0000   Median :2.000   Median :0.0000
##  Mean   :71.95   Mean   :0.7922   Mean   :2.184   Mean   :0.2345
##  3rd Qu.:87.00   3rd Qu.:1.0000   3rd Qu.:3.000   3rd Qu.:0.2500
##  Max.   :94.00   Max.   :3.0000   Max.   :4.000   Max.   :5.0000
##  NA's   :833     NA's   :831      NA's   :341     NA's   :343
```

Podemos observar que es un set de datos de un modelo supervisado pero para fines de nuestra PEC 2 vamos a ejecutar un modelo no supervisado tomando la columna "Species" como la variable que vamos a predecir. Primeramente observamos 3 valores:
CH=Halcón de Cooper RT=Colirrojo SS=Gavilán Podemos observar que hay más datos para Colirrojo y Gavilán. Vamos a ver como se comporta el algoritmo k-means con esta distribución de datos.

k-means

Comenzamos a evaluar el número de cluster que necesitamos para nuestra variable "k"

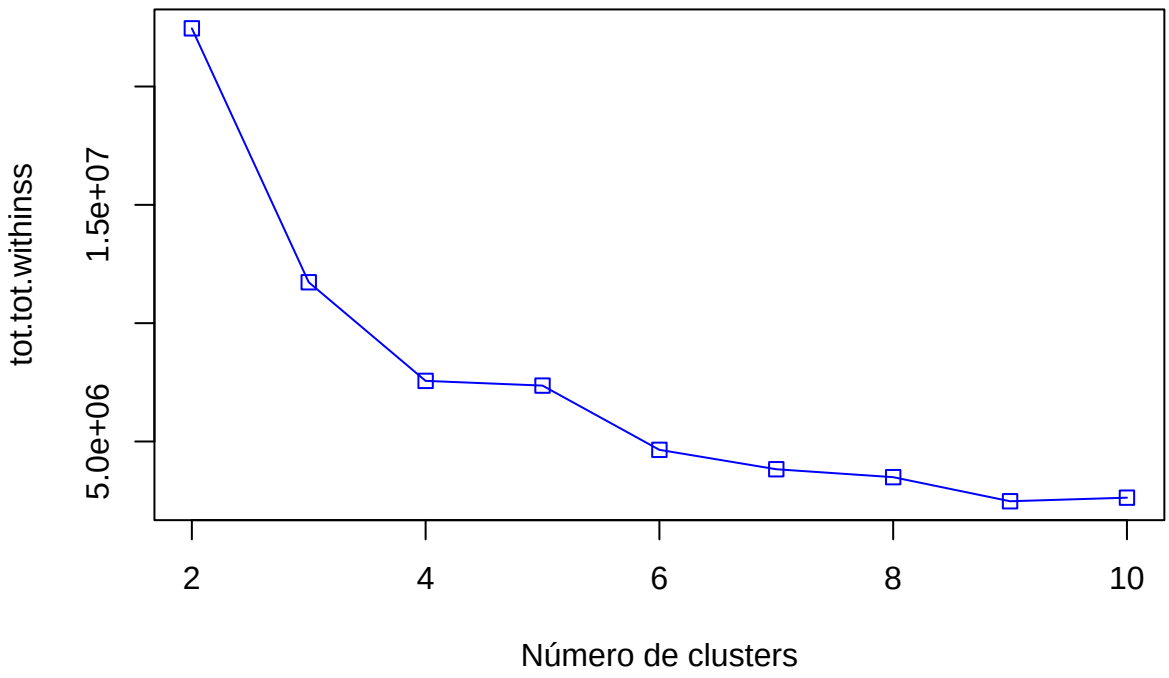
```
distance <- daisy(hawks_k_means)
resultados <- rep(0, 10)
for (i in c(2,3,4,5,6,7,8,9,10))
{
  fit           <- kmeans(hawks_k_means, i)
  y_cluster     <- fit$cluster
  sk            <- silhouette(y_cluster, distance)
  resultados[i] <- mean(sk[,3])
}
plot(2:10,resultados[2:10],type="o",col="blue",pch=0,xlab="Número de clusters",ylab="Silueta")
```



De acuerdo a los valores de las siluetas, el mejor valor para "k" es 2 a pesar que hay 3 tipos de especie.

Vamos a verificar el número de cluster mediante el procedimiento elbow (codo).

```
resultados <- rep(0, 10)
  for (i in c(2,3,4,5,6,7,8,9,10))
  {
    fit           <- kmeans(hawks_k_means, i)
    resultados[i] <- fit$tot.withinss
  }
plot(2:10,resultados[2:10],type="o",col="blue",pch=0,xlab="Número de clusters",ylab="tot.tot.withinss")
```



Como observamos, de acuerdo al método de elbow, el valor mas optimo para "k" podría ser 4 o 6.

Vamos a utilizar los criterios, silueta media ("asw") y Calinski-Harabasz ("ch").

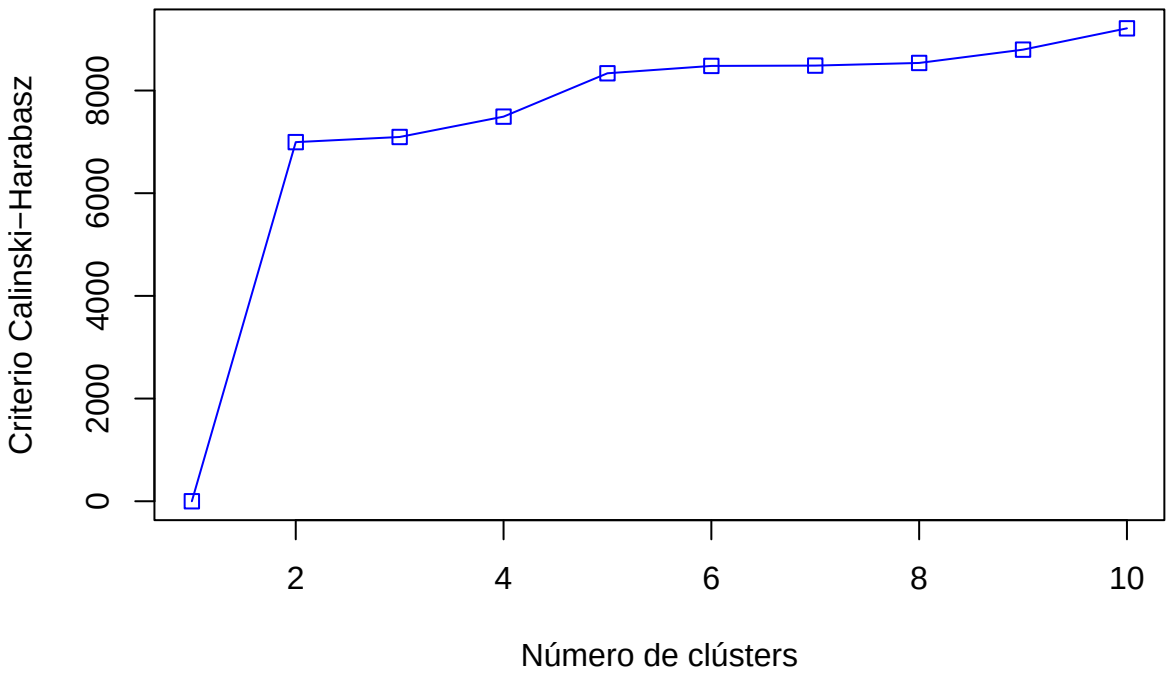
```
if (!require('fpc')) install.packages('fpc')
library(fpc)

fit_ch  <- kmeansruns(hawks_k_means, krange = 1:10, criterion = "ch")
fit_asw <- kmeansruns(hawks_k_means, krange = 1:10, criterion = "asw")

print(fit_ch$bestk)
```

```
## [1] 10
```

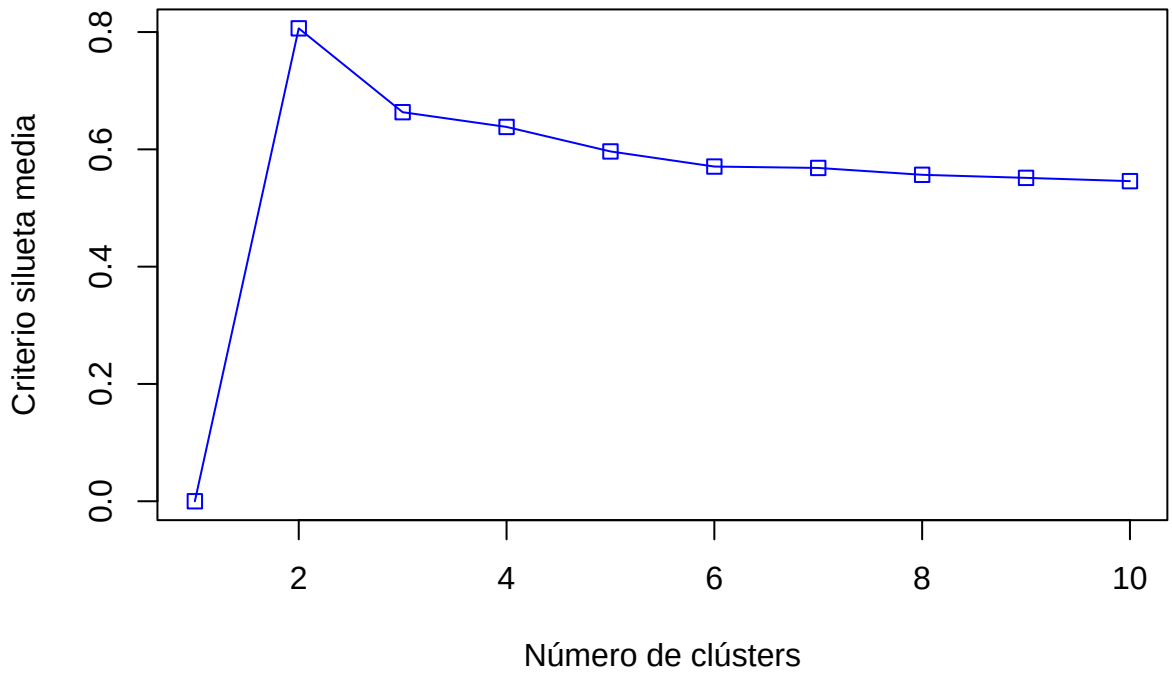
```
print(fit_asw$bestk)
```

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

```
plot(1:10,fit_ch$crit,type="o",col="blue",pch=0,xlab="Número de clústers",ylab="Criterio Calinski-Haraba
```




```
plot(1:10,fit_asw$crit,type="o",col="blue",pch=0,xlab="Número de clústers",ylab="Criterio silueta media
```



De acuerdo a los criterios ch y asw, el número para "k" podría ser 3, este resultado es el mas cercano al número de especies que ya conocemos. PAra fines de nuestra PEC 2, vamos a continuar con el valor de "k" igual a 3.

Clasificación k-means

Aplicamos la función de kmeans para 3 clusters

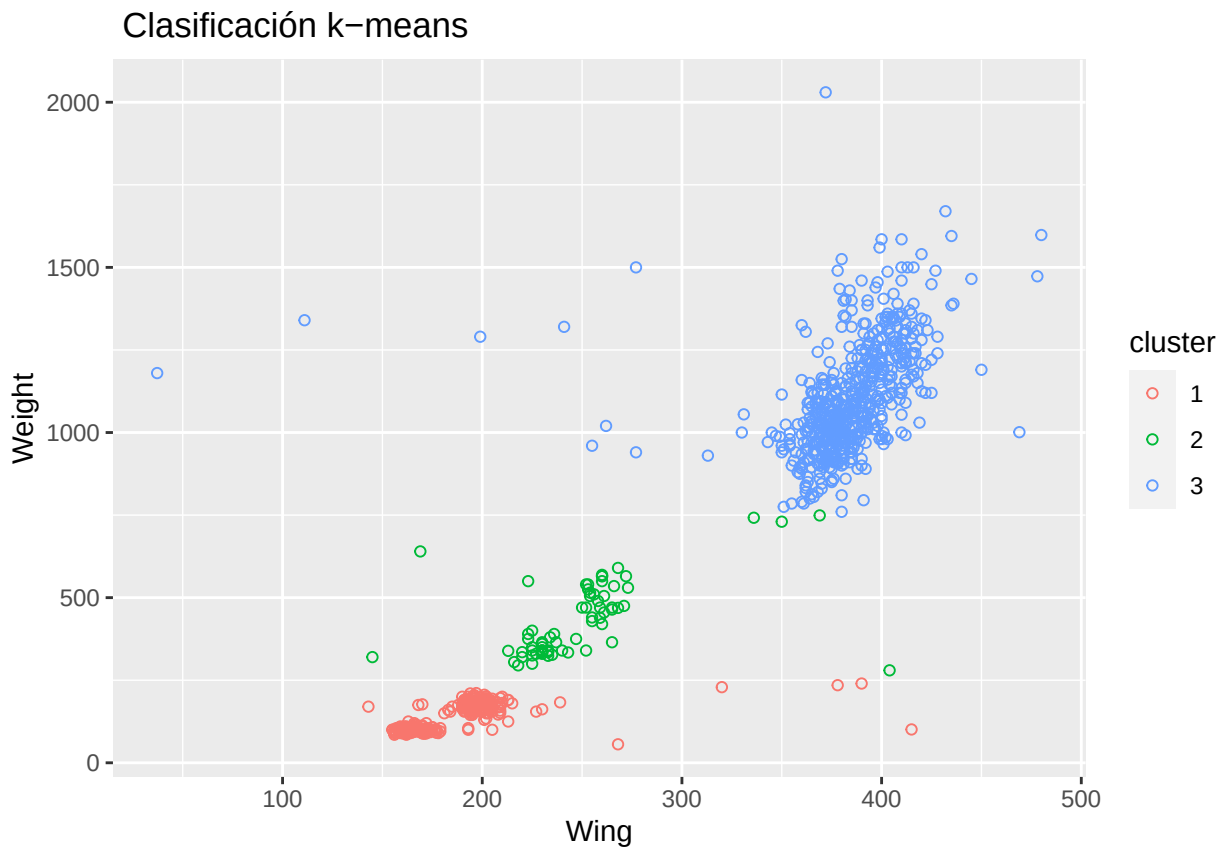
```
hawks3clusters <- kmeans(hawks_k_means, 3)
hawks_k_means$cluster <- as.character(hawks3clusters$cluster)
hawks3clusters
```

```
## K-means clustering with 3 clusters of sizes 260, 69, 562
##
## Cluster means:
##       Wing    Weight   Culmen   Hallux
## 1 187.0615  141.4462 11.58135 15.29212
## 2 248.1449  431.8406 17.94058 22.68333
## 3 383.8989 1104.8683 27.01548 32.01619
##
## Clustering vector:
##   1   3   4   5   6   7   8   9  10  11  12  13  15  16  17  18  19  20  21  22
##   3   3   2   1   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3   3
##  23  24  25  27  28  29  30  31  32  33  34  35  36  37  38  39  40  41  42  43
##   1   3   3   3   3   1   3   3   3   3   3   3   3   3   3   3   1   1   2   3
##  44  45  46  47  48  49  50  51  52  53  54  55  56  57  58  59  60  61  62  63
```

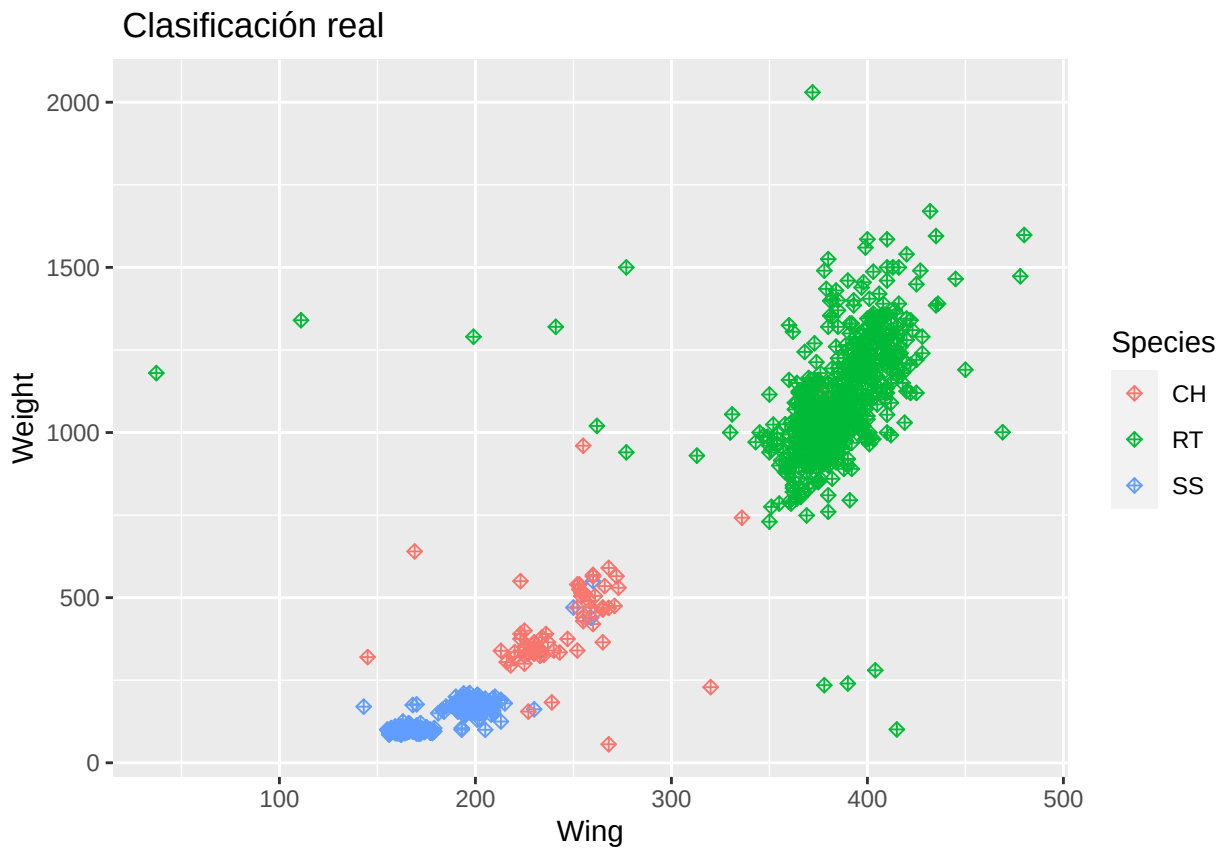
```
##   3   3   3   3   3   3   3   3   3   1   3   3   2   2   3   3   3   3   3   3
##  65  66  67  68  70  71  72  73  74  75  76  77  78  79  80  81  82  83  84  85
##   1   3   2   1   3   3   3   3   3   3   3   3   3   3   3   3   3   1   1   1
##  86  87  88  89  90  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105
##   2   3   3   3   3   3   3   3   3   3   3   3   1   1   1   1   3   1   1   3
## 106 107 108 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125
##   3   1   2   3   3   2   1   3   3   3   3   3   3   3   3   1   1   3   3   1
## 126 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144 145
##   3   3   1   1   1   3   3   3   3   3   3   3   3   3   3   1   1   1   1   3
## 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162 163 164 165
##   3   3   3   3   3   3   1   3   3   3   3   3   3   3   1   1   3   3   3   3
## 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180 181 182 183 184 185
##   3   1   1   3   1   1   1   3   3   3   1   3   3   3   3   3   3   3   3   3
## 186 187 188 189 190 191 192 193 195 196 197 198 199 200 201 202 203 204 205 206
##   3   3   1   3   3   3   3   3   1   1   3   1   3   3   1   1   3   3   3   3
## 207 208 209 210 211 212 213 214 215 216 217 219 220 221 222 223 224 225 226 227
##   3   3   3   3   1   1   1   1   1   1   3   1   3   2   3   3   3   3   3   1
## 228 229 230 231 232 233 234 235 236 237 238 239 240 241 242 243 244 245 246 247
##   1   3   3   1   3   3   2   3   3   3   3   2   3   3   3   3   3   3   3   3
## 248 249 250 251 252 253 254 255 256 257 258 259 260 261 262 264 265 266 267 268
##   3   2   3   3   3   1   3   3   3   3   3   3   3   3   3   3   3   3   3   1
## 269 270 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288
##   3   3   3   2   3   3   3   1   1   3   3   3   3   3   3   1   3   3   2   2
## 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306 307 308
##   3   3   3   3   3   3   3   3   3   3   2   3   1   1   3   1   1   1   1   1
## 309 310 311 312 313 314 316 317 318 319 320 321 322 323 324 325 326 327 328 329
##   3   2   1   1   1   1   1   3   2   1   1   2   3   3   3   2   2   2   1   1
## 330 331 333 334 335 336 337 338 339 340 341 342 343 344 345 346 347 348 349 350
##   3   3   3   3   3   3   3   3   3   1   1   3   3   3   3   3   3   3   3   3
## 351 352 353 354 355 356 357 358 359 360 361 362 363 364 365 366 367 368 369 370
##   3   1   2   3   3   3   1   2   3   1   3   3   1   3   3   3   3   1   3   1
## 371 372 373 374 375 376 377 378 379 380 381 382 383 384 385 386 387 388 389 390
##   2   3   2   1   1   3   3   3   1   3   3   3   1   3   3   1   3   3   3   3
## 391 392 393 394 395 396 397 398 399 400 401 402 403 404 405 406 408 409 410 411
##   3   3   2   1   1   3   3   3   3   2   3   3   3   1   3   3   3   3   3   3
## 412 413 415 416 417 418 419 421 422 423 424 425 426 427 428 429 430 431 432 434
##   3   1   2   1   1   3   1   3   1   3   2   1   3   3   3   1   3   1   1   3
## 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450 451 452 453 454
##   3   3   3   3   1   3   3   2   2   3   2   3   1   1   1   3   1   3   3   1
## 455 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474
##   3   3   3   1   2   3   1   3   3   1   3   3   3   3   3   3   3   3   3   3
## 475 476 477 478 479 480 481 482 483 484 485 486 487 488 489 490 491 492 493 494
##   1   3   1   3   3   3   3   3   3   3   3   1   3   1   3   2   3   3   1   3
## 495 496 497 498 499 500 501 502 503 504 505 506 507 508 509 510 511 512 513 514
##   1   3   3   3   3   1   3   3   1   3   3   3   3   3   1   3   1   3   1   3
## 515 516 517 518 519 520 521 522 524 526 527 528 529 530 531 532 533 534 535 536
##   3   1   3   1   1   1   2   3   2   3   1   3   3   2   3   3   2   3   1   3
## 537 538 539 540 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556
##   3   3   3   3   3   2   3   3   1   1   3   3   3   3   3   3   3   3   3   3
## 557 558 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576
##   1   3   3   3   3   1   1   3   1   3   3   1   1   1   3   3   3   1   1   3
## 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594 595 596
##   2   3   3   1   3   3   1   3   3   3   1   2   3   3   3   3   3   3   1   3
## 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614 615 616
```

```
##   3   3   2   3   3   3   3   1   3   3   3   3   3   3   1   3   3   3   3   3
## 617 618 619 620 621 622 623 624 625 626 627 628 629 630 631 632 633 634 635 636
##   3   1   1   3   1   3   1   1   1   1   1   1   3   1   3   3   1   2   1   1
## 637 638 639 640 641 642 643 644 645 646 647 648 649 650 651 652 653 654 655 656
##   1   1   1   1   3   1   2   1   1   1   3   3   3   3   3   3   1   1   1   1
## 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671 672 673 674 675 676
##   1   3   1   1   1   1   1   3   3   3   3   1   3   1   3   3   2   3   1   3
## 677 678 679 680 681 682 683 684 685 686 687 688 689 690 691 692 693 694 695 696
##   1   3   3   2   3   3   3   1   1   2   1   1   1   1   3   3   1   3   3   3
## 697 698 699 700 701 702 703 704 705 706 707 708 709 710 711 712 713 714 715 716
##   1   1   3   3   3   2   1   3   3   3   3   2   3   1   2   1   1   3   3   3
## 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736
##   3   3   3   3   3   1   3   1   1   3   1   3   1   1   3   3   3   1   1   1
## 737 738 739 740 741 742 743 744 745 746 747 748 749 750 751 752 753 754 755 756
##   1   1   1   3   2   2   3   3   3   1   1   1   1   3   1   3   1   2   1   1
## 757 758 759 760 761 762 763 764 765 766 767 768 769 770 771 772 773 774 775 776
##   2   3   3   3   1   3   1   3   3   3   3   3   3   3   3   2   3   3   3   1
## 777 778 779 780 781 782 783 784 785 786 787 788 789 790 791 792 793 794 795 796
##   3   3   1   3   1   3   1   3   3   3   3   2   3   3   1   3   3   1   3   3
## 797 798 799 800 801 802 803 804 805 806 807 808 809 810 811 812 813 814 815 816
##   3   3   1   3   1   1   1   3   1   1   1   2   2   1   3   3   1   1   1   3
## 817 818 819 820 821 822 823 824 825 826 827 828 829 830 831 832 833 834 835 836
##   3   1   2   1   2   1   3   3   1   2   3   1   1   2   1   3   3   1   1   1
## 837 838 839 840 841 842 843 844 845 846 847 848 849 850 851 852 853 854 855 856
##   1   3   2   1   1   2   3   3   3   3   3   3   1   2   2   1   3   3   3   3
## 857 858 859 860 861 862 863 864 865 866 867 868 869 870 871 872 873 874 875 876
##   1   3   1   3   3   3   3   1   1   3   3   3   3   1   3   3   1   1   3   3
## 877 878 879 880 881 882 883 884 885 887 888 889 890 891 892 893 894 895 896 897
##   2   3   3   3   3   3   3   1   3   1   3   2   1   3   3   3   3   1   1   1
## 898 899 900 901 902 903 904 905 906 907 908
##   3   1   3   3   3   3   3   1   3   3   3
##
## Within cluster sum of squares by cluster:
## [1]   729507.3   955099.3 16512800.6
##  (between_SS / total_SS =  90.9 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"     "tot.withinss"
## [6] "betweenss"    "size"         "iter"         "ifault"
```

```
#Wing and Weight
#plot(hawks_k_means[c(1,2)], col=hawks3clusters$cluster, main="Clasificación k-means")
ggplot(hawks_k_means) + geom_point(aes(x=Wing, y=Weight, colour=cluster), shape=1) + labs(title= " Clas
```

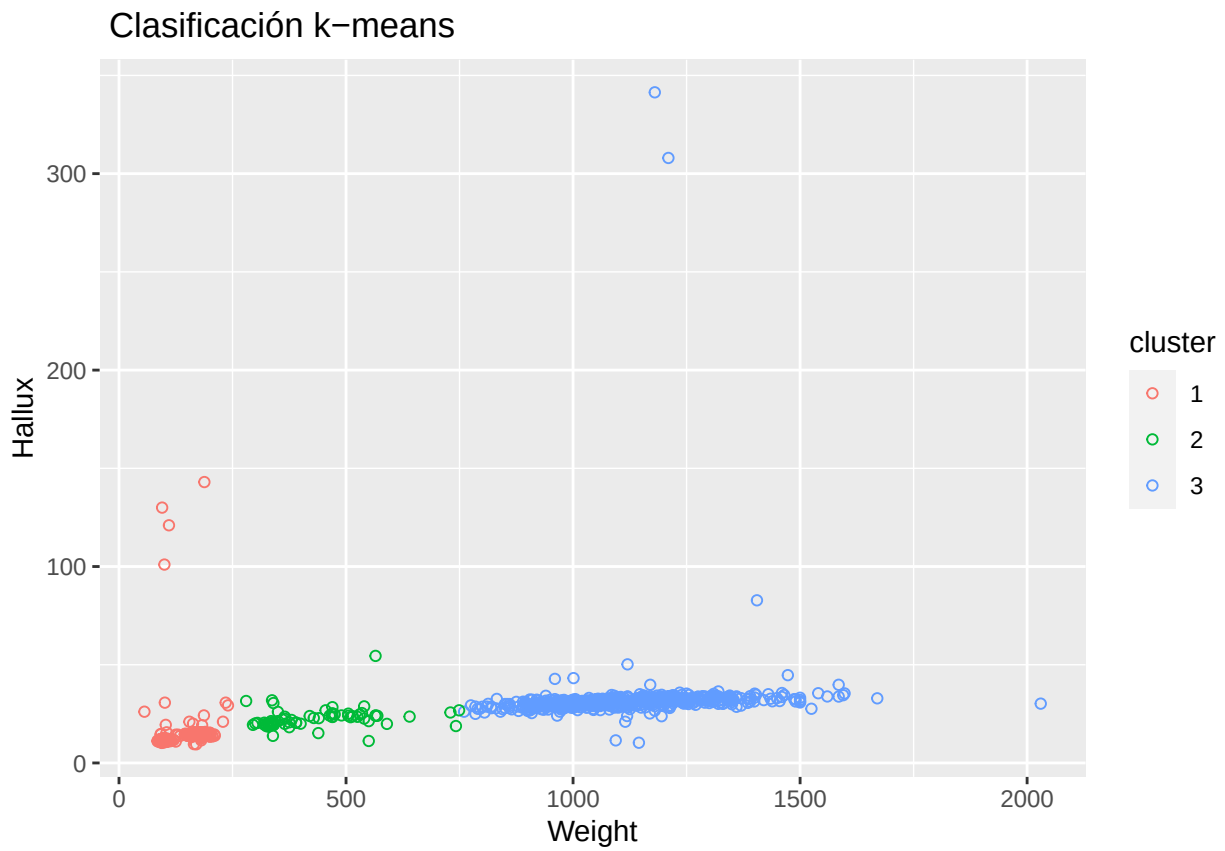



```
#Wing and Weight
ggplot(hawks_k_original) + geom_point(aes(x=Wing, y=Weight, colour=Species), shape=9)+ labs(title= " Cl
```

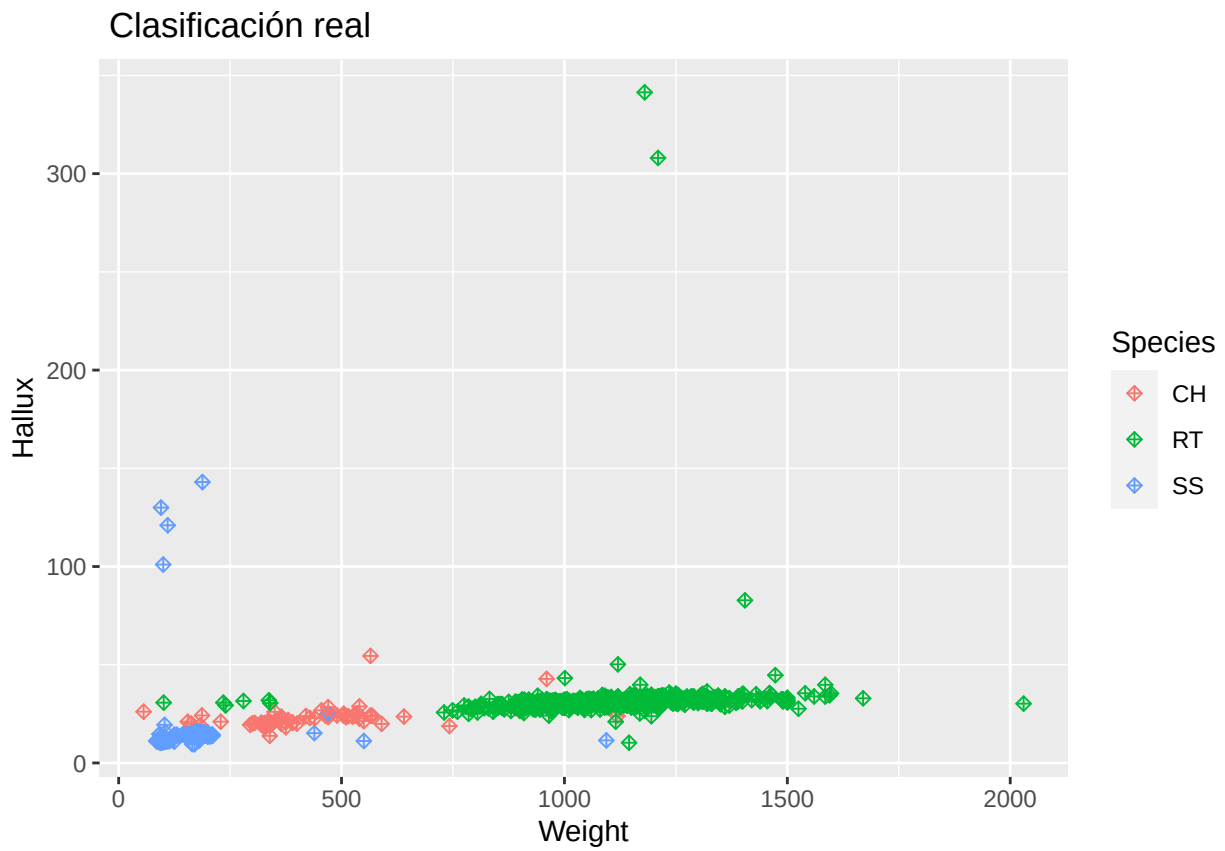



```
#plot(hawks_k_original[c(4,5)], col=as.factor(hawks_k_original$Species), main="Clasificación real")
```

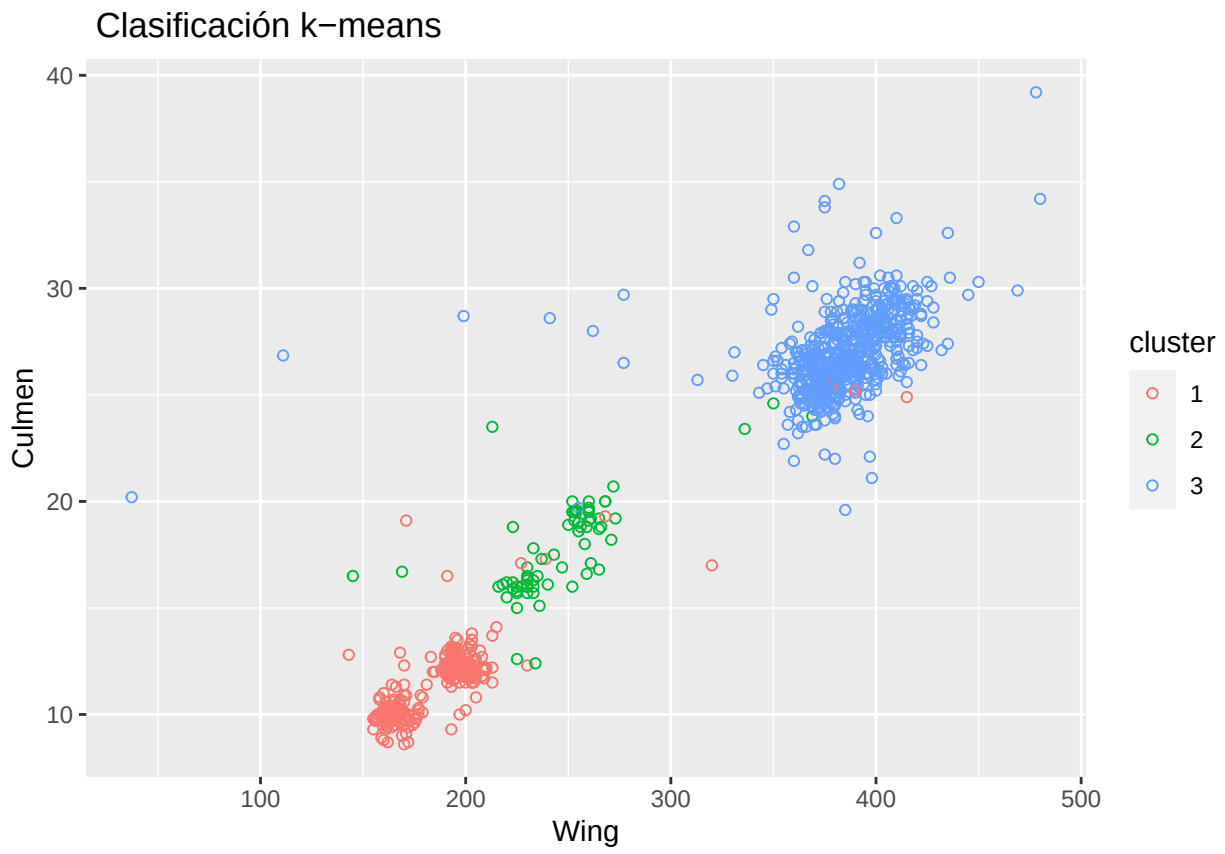
```
#Hallux and Weight
#plot(hawks_k_means[c(2,4)], col=hawks3clusters$cluster, main="Clasificación k-means")
ggplot(hawks_k_means) + geom_point(aes(x=Weight, y=Hallux, colour=cluster), shape=1) + labs(title= " Cl
```



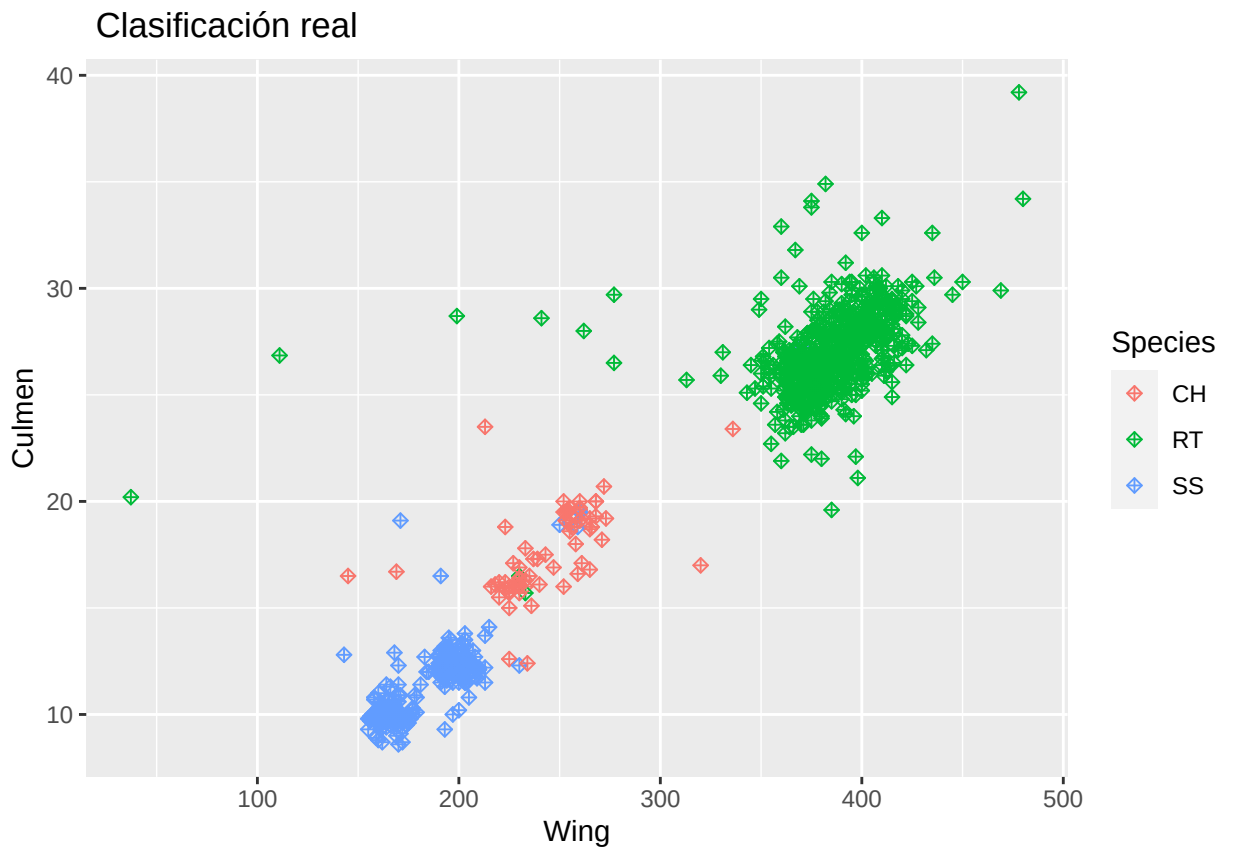

```
#Hallux and Weight
#plot(hawks_k_original[c(2,4)], col=as.factor(hawks_k_original$Species), main="Clasificación real")
ggplot(hawks_k_original) + geom_point(aes(x=Weight, y=Hallux, colour=Species), shape=9)+ labs(title= " (
```




```
#Culnmen and Wing
#plot(hawks_k_means[c(1,3)], col=hawks3clusters$cluster, main="Clasificación k-means")
ggplot(hawks_k_means) + geom_point(aes(x=Wing, y=Culmen, colour=cluster), shape=1) + labs(title= " Clas
```



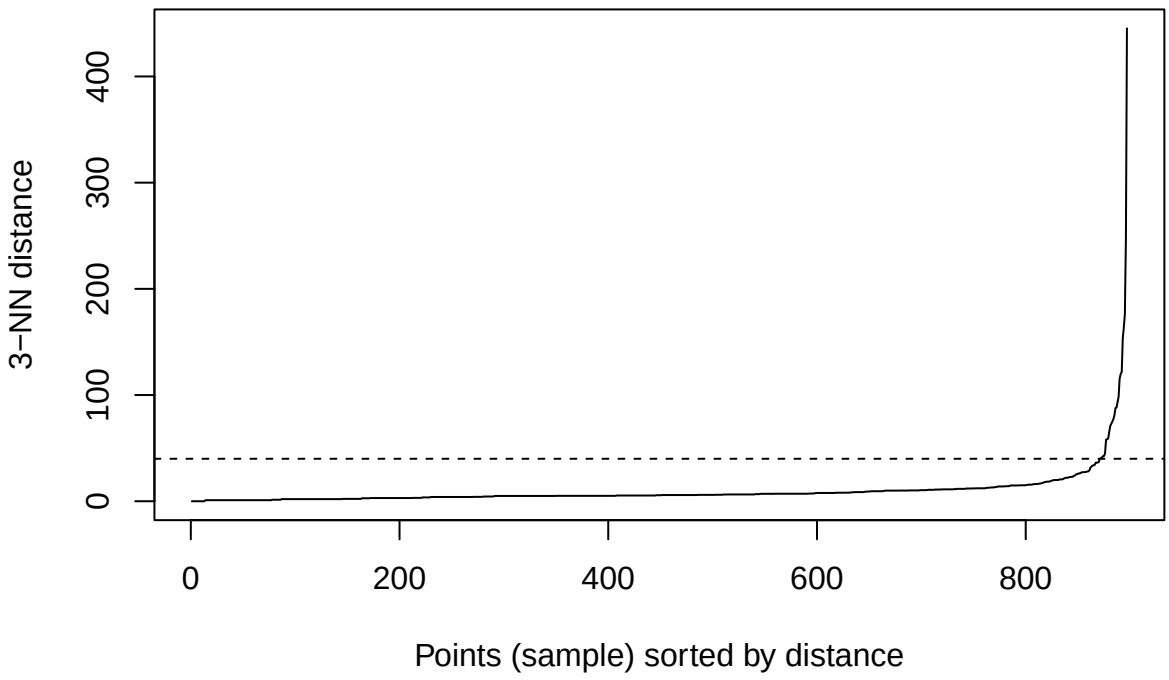

```
#Culnmen and Wing
#plot(hawks_k_original[c(1,3)], col=as.factor(hawks_k_original$Species), main="Clasificación real")
ggplot(hawks_k_original) + geom_point(aes(x=Wing, y=Culmen, colour=Species), shape=9)+ labs(title= " Cla
```



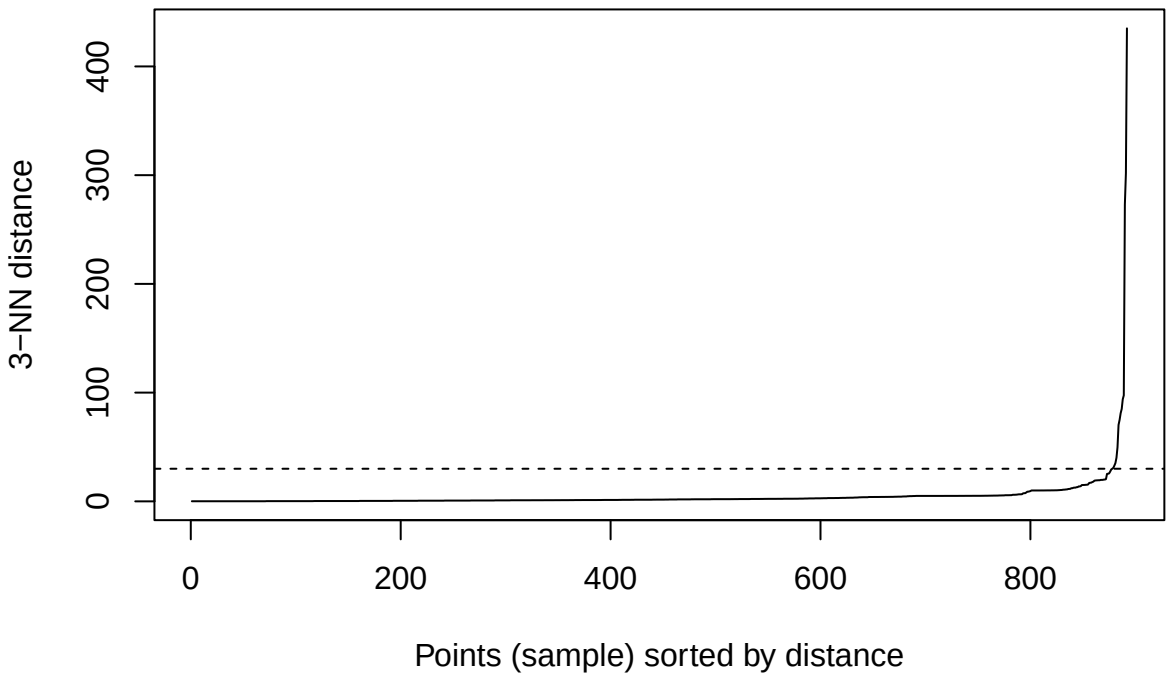
En todas las gráficas logramos ver los 3 cluster definidos, resalto el gráfico Hallux y Weight, donde la distribución de los clusters de kmeans con respecto al gráfico real, tienen mas similitud que con los graficos restantes.

dbscan

```
eps_plot = kNNdistplot(hawks_k_means_dbscan_wing_weight, k=3)
eps_plot %>% abline(h = 40, lty = 2)
```



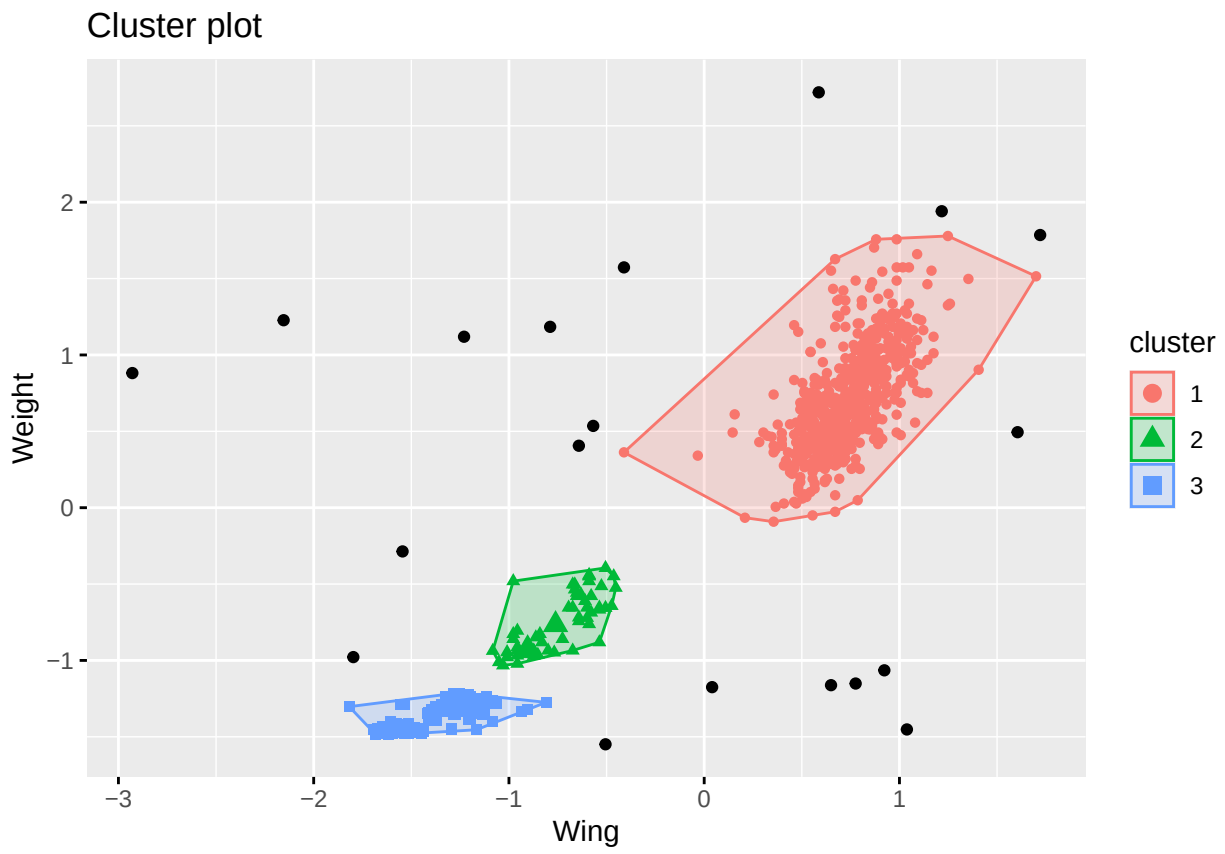

```
eps_plot = kNNdistplot(hawks_k_means_dbscan_hallux_weight, k=3)
eps_plot %>% abline(h = 30, lty = 2)
```




```
wing_weight <- dbscan::dbscan(hawks_k_means_dbscan_wing_weight, eps = 50, MinPts =  5)
```

```
## Warning in dbscan::dbscan(hawks_k_means_dbscan_wing_weight, eps = 50, MinPts =
## 5): converting argument MinPts (fpc) to minPts (dbscan)!
```

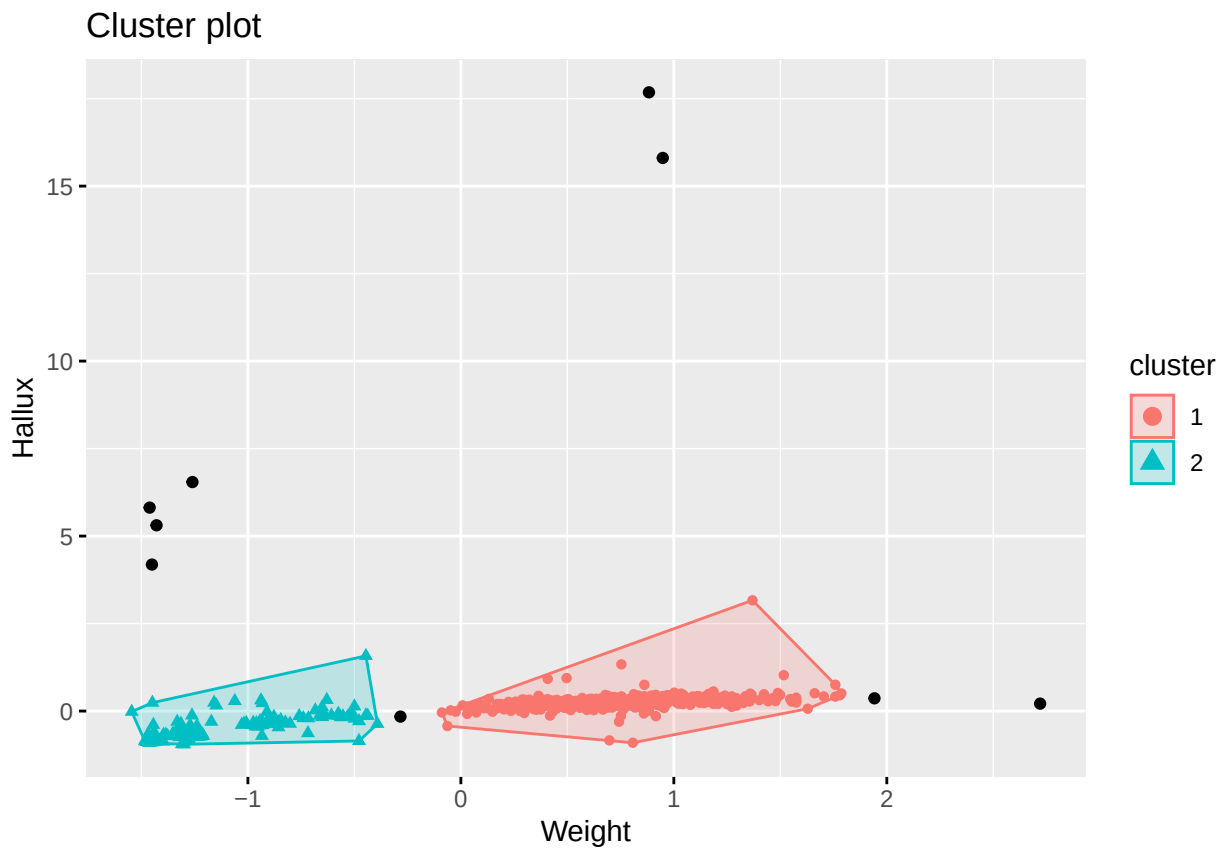
```
fviz_cluster(wing_weight, hawks_k_means_dbscan_wing_weight, geom = "point")
```




```
hallux_weight <- dbscan::dbscan(hawks_k_means_dbscan_hallux_weight, eps = 50, MinPts =  5)
```

```
## Warning in dbscan::dbscan(hawks_k_means_dbscan_hallux_weight, eps = 50, :
## converting argument MinPts (fpc) to minPts (dbscan)!
```

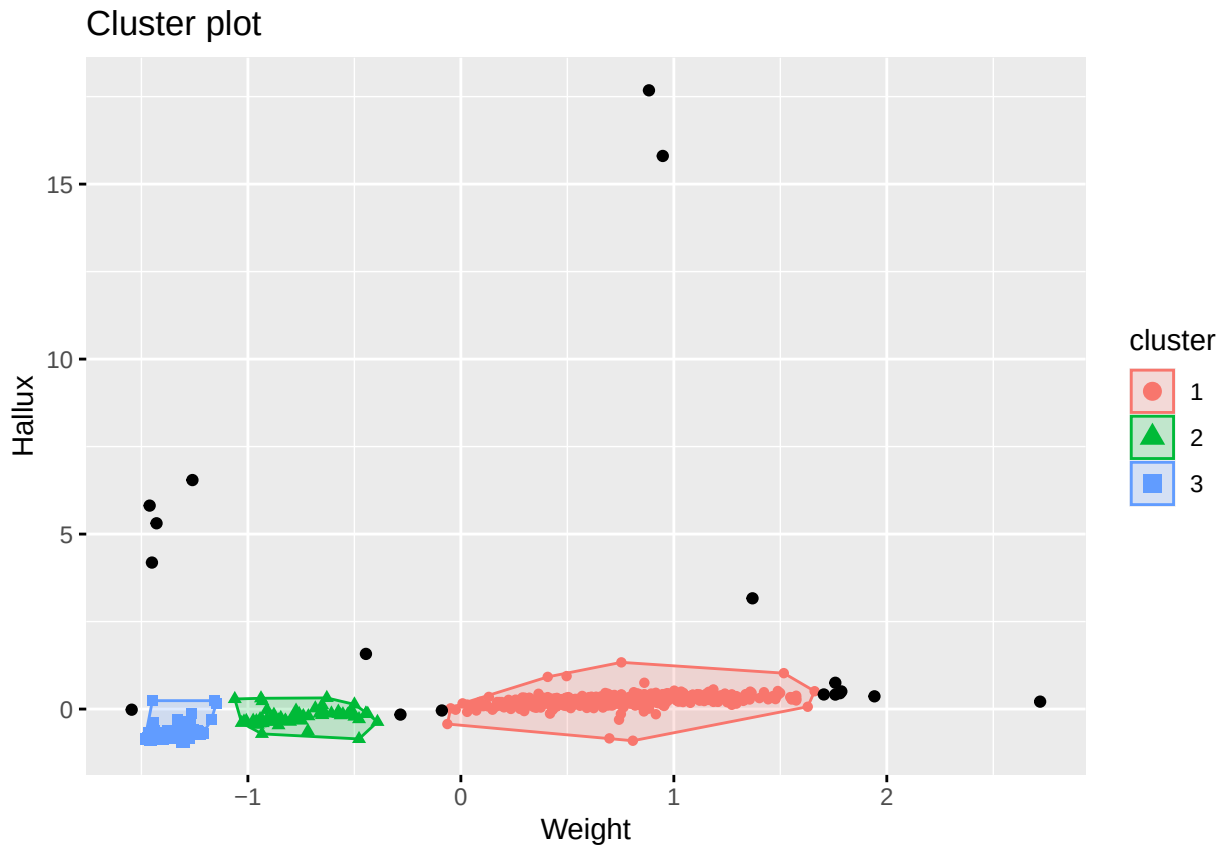
```
fviz_cluster(hallux_weight, hawks_k_means_dbscan_hallux_weight, geom = "point")
```




```
hallux_weight <- dbscan::dbscan(hawks_k_means_dbscan_hallux_weight, eps = 30, MinPts =  6)
```

```
## Warning in dbscan::dbscan(hawks_k_means_dbscan_hallux_weight, eps = 30, :
## converting argument MinPts (fpc) to minPts (dbscan)!
```

```
fviz_cluster(hallux_weight, hawks_k_means_dbscan_hallux_weight, geom = "point")
```



##OPTICS

```
hawks_k_means_dbscan_wing_weight_bk <- hawks_k_means_dbscan_wing_weight
```

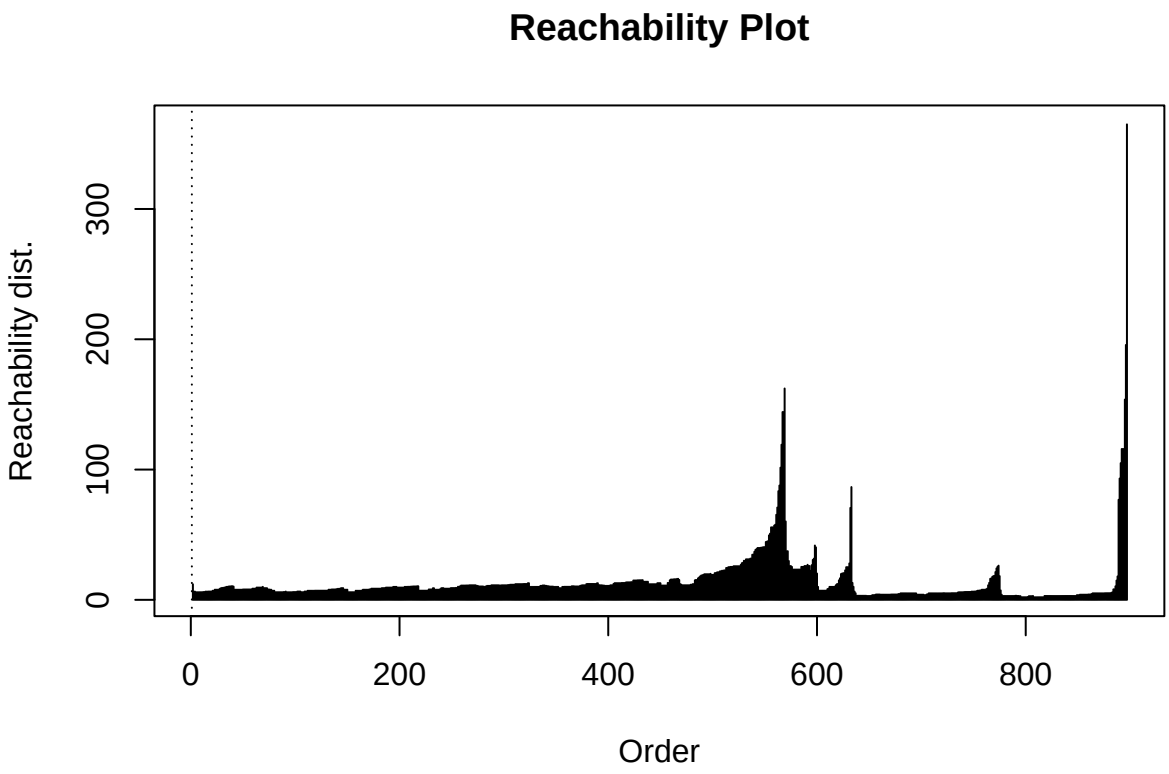
```
hawks_k_means_dbscan_wing_weight <- hawks_k_means_dbscan_wing_weight_bk
#hawks_k_means_dbscan_wing_weight <- scale(hawks_k_means_dbscan_wing_weight)
```

```
opctics_wing_weight <- optics(hawks_k_means_dbscan_wing_weight, minPts = 10)
opctics_wing_weight$order
```

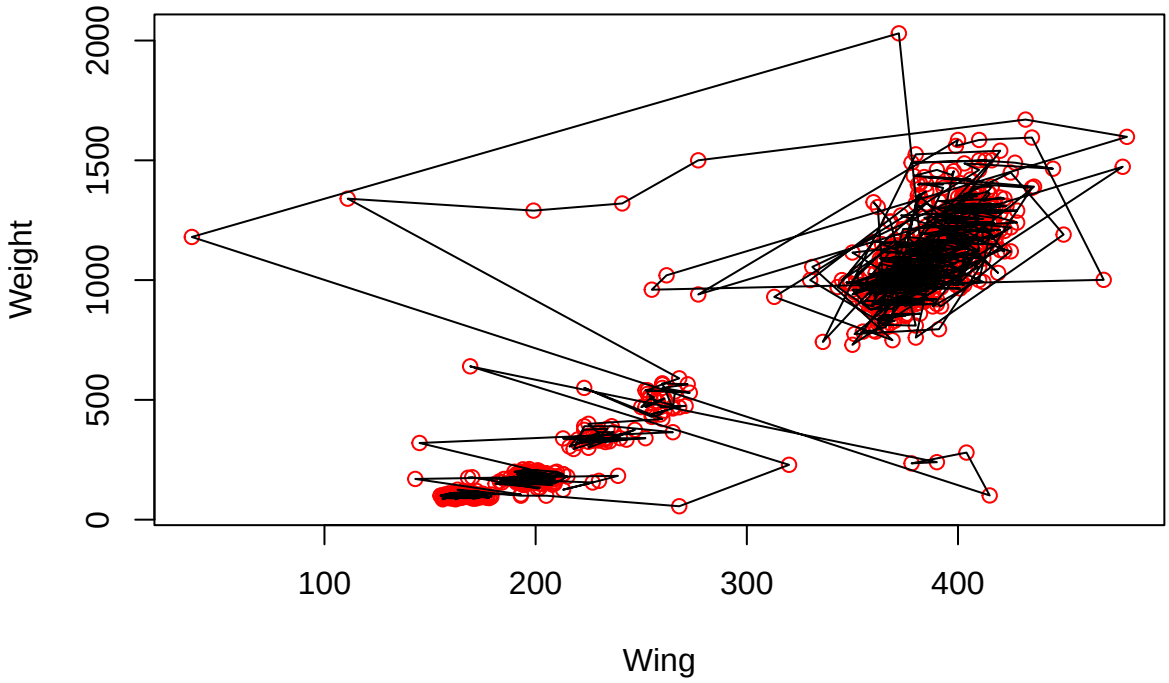
```
##   [1]   1 835 561 469 856 715 533 374 290 262 254 545 358 226  77   2 661 549
##  [19] 485  18 317 218 547 461 252 103 257 565 203 733 548 602 457 428 150 762
##  [37] 277 243 867 585 816 604 568 560 281 256 153 136  17 499 890 875 318 247
##  [55] 696 550 536 515 851 833 431   7 668 722 147 146 132 384 601 821 789 670
##  [73] 637 704 570 778 690 571 759 609 594 506 330 120 116  90  24   3 521 417
##  [91] 860 753 734 478 474 181 110 850 844 200  12 812 767 865 639 472 458 401
## [109] 289 152 115 813 754 590 496 462 379 241 887 880 583 539 418 328 311 265
## [127] 182 134 128  89  76 464 162 113 363 278 276  45 232 414 273 219 747 729
## [145] 638 497 732 451 837 596 578 456 372 287 246 143  47 776 751 655 582 196
## [163] 187 337 544 448 280 190  87  46 760 348 441 359 229 202 177 763 366 156
## [181]  84  42 881 597 131 331 877 801 706 606 429 292  44 882 774 520 495 749
## [199] 269 180 178 175  19  11 421 332  30 505 396 385 176 151 490 236 191  32
## [217] 843 636 618 595 555 541 446 424 397 194  67 773 660 591 589 685 336 781
## [235] 771 339 333 324 579 144 587 399 784 426 179 174 163  26 166 708 681 349
## [253]  35 656 160 492 148 517 620 454 260  99  48 361 214 197 764 698 523 480
## [271] 425 386 261 316 133 834 598 542  85  68 135  13 170 870 603 295  43 758
```

```
## [289] 487 360 237 872 444 350 341 868 832 207 206 205 154 114  10 338 285 540
## [307] 488 476 220  20 871 785 529 453 304 233   6  50 805 574 155  69 720 680
## [325] 787 757 647 630 611 435 419 404 121  74 721 405 705 580 513 800 836 599
## [343] 530 712 694 553 503 483 473 437 412 345 508 253  72 137  88 861 707 641
## [361] 527 117  94   9 382 371 204 621 581 562 378  27 537 605 493 284 142  14
## [379] 259 654 822 402 228 225 222 221 183 171 344 855 806 864 766 739 695 376
## [397] 231 189  62  57 242 671  79  53 793 525 432 286 270 268 255 107 779 518
## [415]  21 340 775 460 393 391 111 874 683 769 201  54  16 343 342 693 159  51
## [433] 741 858 845 264 258 250 748 672 575 329 244 216  92  49 567 427 275 251
## [451] 532 471 248 130  64  59 717  33 827 573 528 410 755  93  36 234 586 891
## [469] 849 556 447 383 381 266 239 184 161 123 124  78 288  70  86 782  23 106
## [487] 370 665 475 640 756 896 857 842 443 112   8 663 883 847 325 238 852 395
## [505] 526  31 658  91 298 102 327 326 889 703 892 463 895 684 501 459 786 538
## [523] 403 353 186  71 710 482  15 543 293 653 470 400  73  61 145 390 498 367
## [541] 240 465 375 172  29  75  34  37  38 320 893 291 267 355 129 869 489 467
## [559] 689 356 709 592 392 188 688  58 897 274 436 730 691 675 777 798 669 662
## [577] 828 577 531 450 347 245  65   4 566 433 235 315 519 394 305 808 700 321
## [595] 407 230 105 294 866 761 839 831 819 697 632 415 319 312  83 365  56 840
## [613] 522 514 282  41 387 352  55 878 623 434 797 588 731 815 283 810 746 217
## [631] 108 512 158 885 830 884 886 888 873 788 740 676 468 398 783 752 322 817
## [649] 765 334 100 307 674 642 195 301 768 738 723 666 724 633 535 509 422 406
## [667] 369 272 167  28 664 649 608 584 477 438 335 796 727 809 829 894 791 736
## [685] 735 716 825 790 772 711 677 876 824 792 714 619 610 466 445 430 416 313
## [703] 811 408 308 686 673 624 614 613 600 546 510 507 504 439 211 208 168 157
## [721] 628 452 719 455 310 300 215 101 389 296 279 165  81   5  95 271 199 173
## [739] 140  80 364 558 557 449 377 368 223 770 678 198 612 491 169 479 164 249
## [757] 126 846 682 651 192 803  96 814  52 224 346 354 631 213 351 185 853 879
## [775] 484 745 728 841 750 742 702 659 652 648 635 615 554 551 409 362 826 863
## [793] 854 795 726 718 622 593 413 411 306 303 380 309 440 297 862 859 848 820
## [811] 818 807 645 643 626 617 494 744 644 679 687 650 616 607 576 564 559 534
## [829] 486 420 388 357 323 302 802 799 725 701 511 500 314 299 149 122 104  98
## [847]  63  25 625 524 125 823 737 563 263 193 141 118  82 210 127  66 139 838
## [865] 804 713 699 646 552 442 227 119 109  97  40 794 780 627 572 516 373 138
## [883] 209 657 569 423  39 212 502 629 743 692 634 481  22  60 667
```

```
plot(opctics_wing_weight)
```




```
plot(hawks_k_means_dbscan_wing_weight, col = "red")
polygon(hawks_k_means_dbscan_wing_weight[opctics_wing_weight$order,])
```

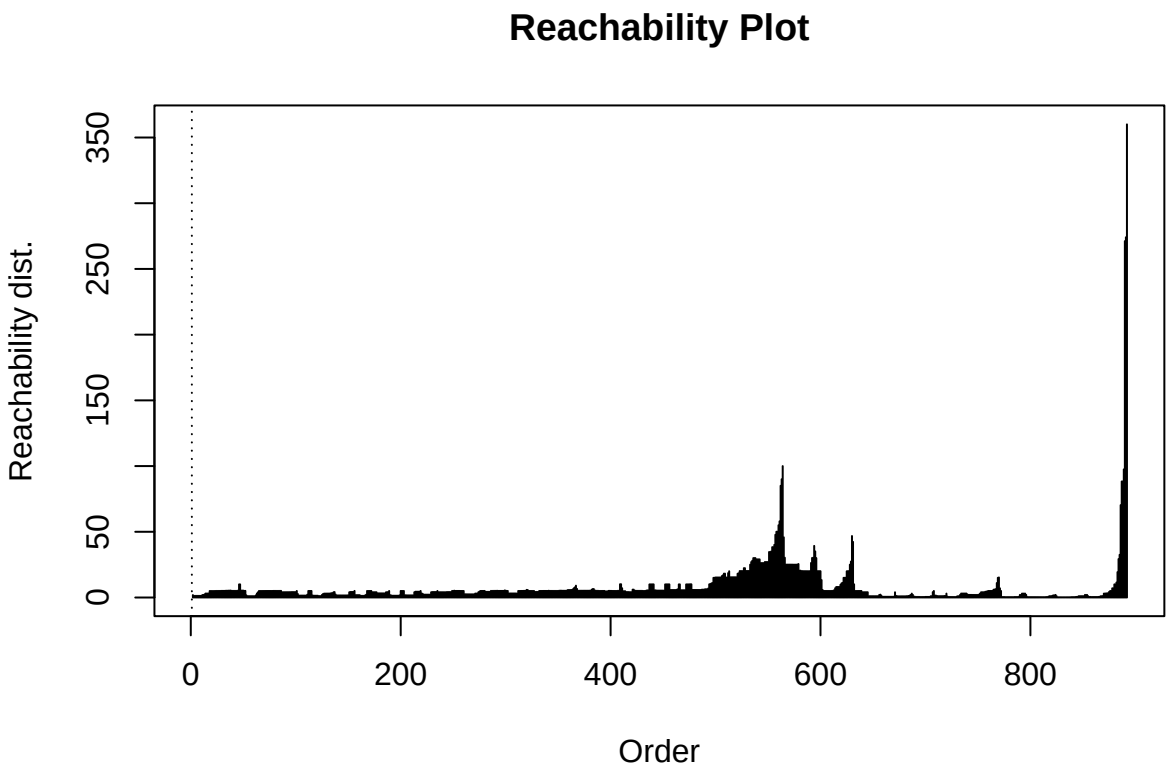



```
opctics_hallux_weight <- optics(hawks_k_means_dbscan_hallux_weight, minPts = 10)
opctics_hallux_weight$order
```

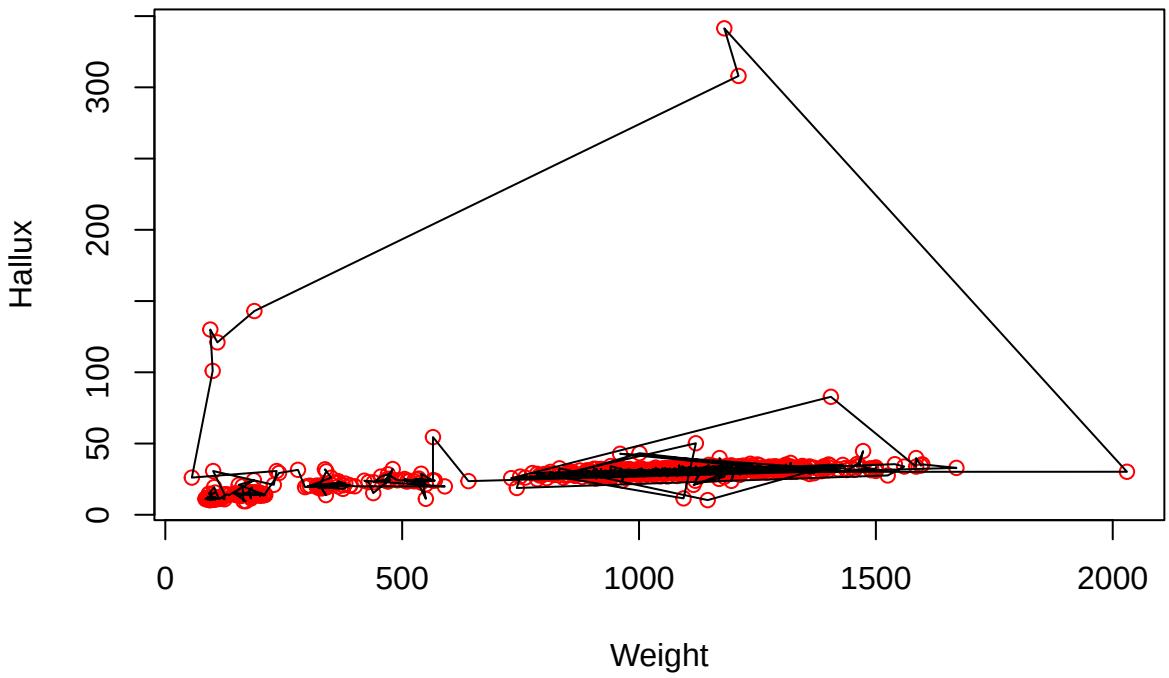
```
##   [1]   1 557 541 529 465 371 288 251 215 200 711 254 315 852 561 544 355 729
##  [19] 657 545 249  17 831 481 260 598 543 101 457 275 240 203  75 237 223 453
##  [37] 148 863 694 513 424 421 407 202 266 198 839 758 581 812 885 692 600 564
##  [55] 546 511 358 279 244 134  16 516 316 495 532 145 718 765 664 144   6 847
##  [73] 829 570 427 556 127 253 679 486 449 217 151 130 597 393 817 785 666 633
##  [91] 381 339 111 360 325 700 566 492 415 132 376 774 287 283 238 180 112 109
## [109]  87  48 150 113 414 388 870 875 809 567 517 118 114 105  23   2 390 328
## [127] 460 337 686 458 263 126 160 755 605 586 335 502 276 268 882 590 258 235
## [145] 231 229 179  88  43 327 329 856 730 602 470 846 840 808 749 474 197 178
## [163] 173  11  10 108  18 491 763 750 635 468 861 745 588 398 309 579 535 501
## [181] 302 271 267 216  74 274 454 176 636 743 725 702 634 411 226 188  42  30
## [199] 872 493 129 797 728 833 447 382 369 345 243 233  45  28 418 756 592 574
## [217] 452 141 334 877 772 540 444 285 278 193  85  44 383 747 651 578 494 185
## [235] 356 593 437 425 290 199 175 154  40 363 876 770 759  82 152  19 677 704
## [253] 656 537 394 632 769 614 591 587 767 585 575 551 488 333 191 142   5 442
## [271] 681 336 161 583 777  97 780 396 158 346 174 164 149  65  33 330  24 211
## [289] 177 450 422 652 760 204 194 172 146  46 867 751 616 525 524 521 519 476
## [307] 230 153  49  12   9 131  83 599 771 157 314 830  66 594 866 538 536 754
## [325] 483 472 293 868 168 133  41 252 357 234  15 828 801 514 484 338 717 864
## [343] 626 456 416 402 347 340 219  67 716 676 643 783 431 282 119  13 401 775
## [361] 753 576 607  72 781  58 701 558 617 601 577 489 379 201 143  25  20 509
## [379] 375 523 368 796 832 504 703 637 526 857 708 690 533 499 409 342 469 257
```

```
## [397] 250 367 115 595 479 140 135  92  70   8 549 433  52 650 818 399 341 225
## [415] 324 222 218 181 169 104 187 851 762 789 691 373 284 239 228  77 802  60
## [433]  55 735 860 667 569 428  51 737 854 892 841 823 744 326 262 255 247 213
## [451] 571  90 241  31  47 699 668 563 528 423 128  62 248 273 752 713 467 245
## [469]  57  56  34 884 778  22 286 582 443 886 845 552 380 272 264 121  84  68
## [487] 471 392 378 182 159  76 236 122 705 848 353 689 661 891 853 838 439 110
## [505]   7 259 322 878 659 843 323 440 455 782 654 534 400  89  69  14 350 291
## [523] 522 296 100  29 890 887 706 680 497 459 478 539  59 649 397 387 466 461
## [541] 352 170 372  71  35  27 684  73  36  32 364 318 888 485 289 265 184 865
## [559] 463 685 389 186 739 726 687 671 773 665 562 515 429 313 232 794 573 227
## [577] 446 391 658 824 696 804 527 344 303 256 242  63   3 319 404 103 432 292
## [595] 742 862 727 619 584 510 757 836 835 827 815 628 412 384  81  54  53 693
## [613] 349 362 873 518 508  39 310 317 280 793 106 430 811 281 214 806 477 688
## [631] 842 880 766 678 554 553 445 365 361 826 196 162 138  78 792 879 764 761
## [649] 734 723 719 647 192 189 670 813 305 869 883 784 881 779 748 736 672 596
## [667] 464 331 638 395 386 731 720 707 629 531 505 419 366 165  26 662 299 246
## [685] 270 332 343 825 805 889 787 732 660 609 604 580 500 473 434 645 542 506
## [703] 503 351 403 163 183 786 849 821 768 712 669 610 435 171   4 155  93 615
## [721] 871 820 788 710 673 606 462 426 208 441 306 166 348 807 799 682 405 205
## [739] 308 374 715 624 475 451 448 298 294 212  99 167  79 124 810 277 269 220
## [757] 620 413 311 608 487  94 674 195 156 627 320  50 221 776 790 733 746 837
## [775] 819 741 698 648 644 631 611 547 206 406 724 655 550 190 738 480  95 709
## [793] 642 438 359 859 822 722 850 834 791 555 436 714 410 408 304 147  64 695
## [811] 618 589 548 530 740 209  61 125 385 207 295 307 301 377 844 816 641 640
## [829] 639 622 612 814 683 572 560 490 417 136 855 646 482 613 675 803 858 297
## [847]  96 120 117 102 139 354 300 107 798 795 721 697 621 520 507 496 261  80
## [865] 224 559 116 137  38 512 800 874 321 312 123 370 623 653 565 420 210  21
## [883] 625 630 498  37 603 568  98  86  91 663
```

```
plot(opctics_hallux_weight)
```




```
plot(hawks_k_means_dbscan_hallux_weight, col = "red")
polygon(hawks_k_means_dbscan_hallux_weight[opctics_hallux_weight$order,])
```

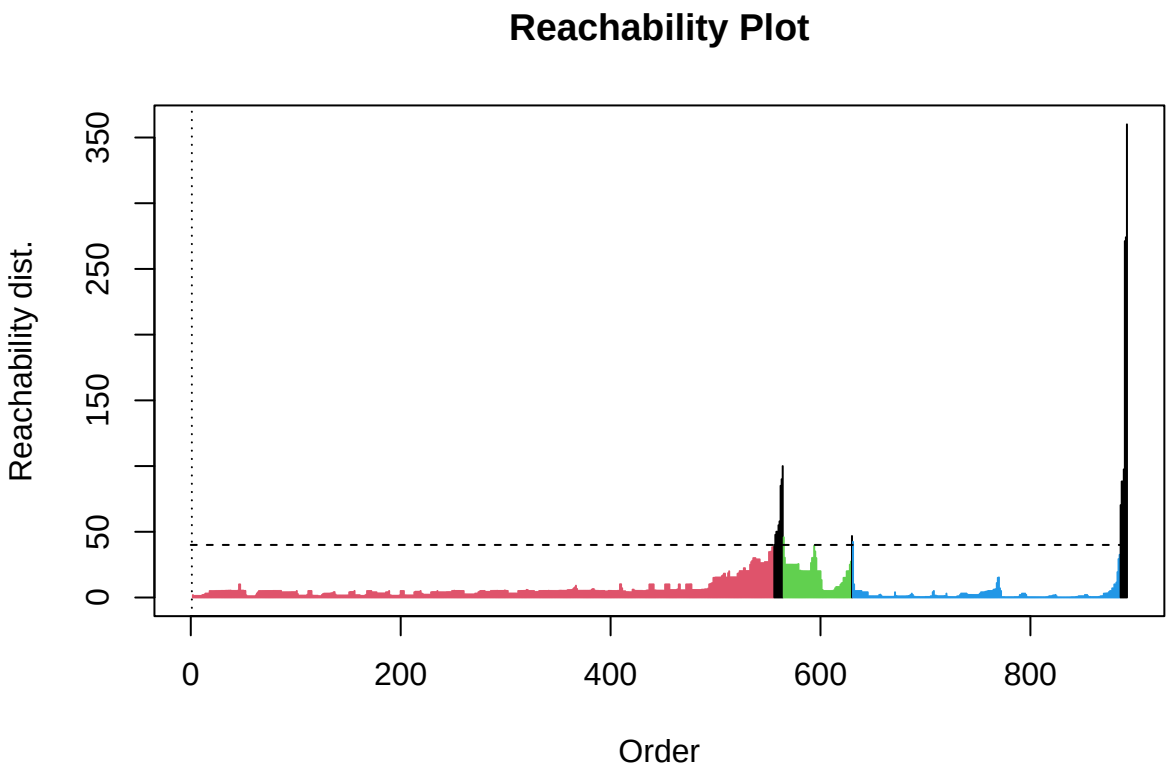


##extractDBSCAN

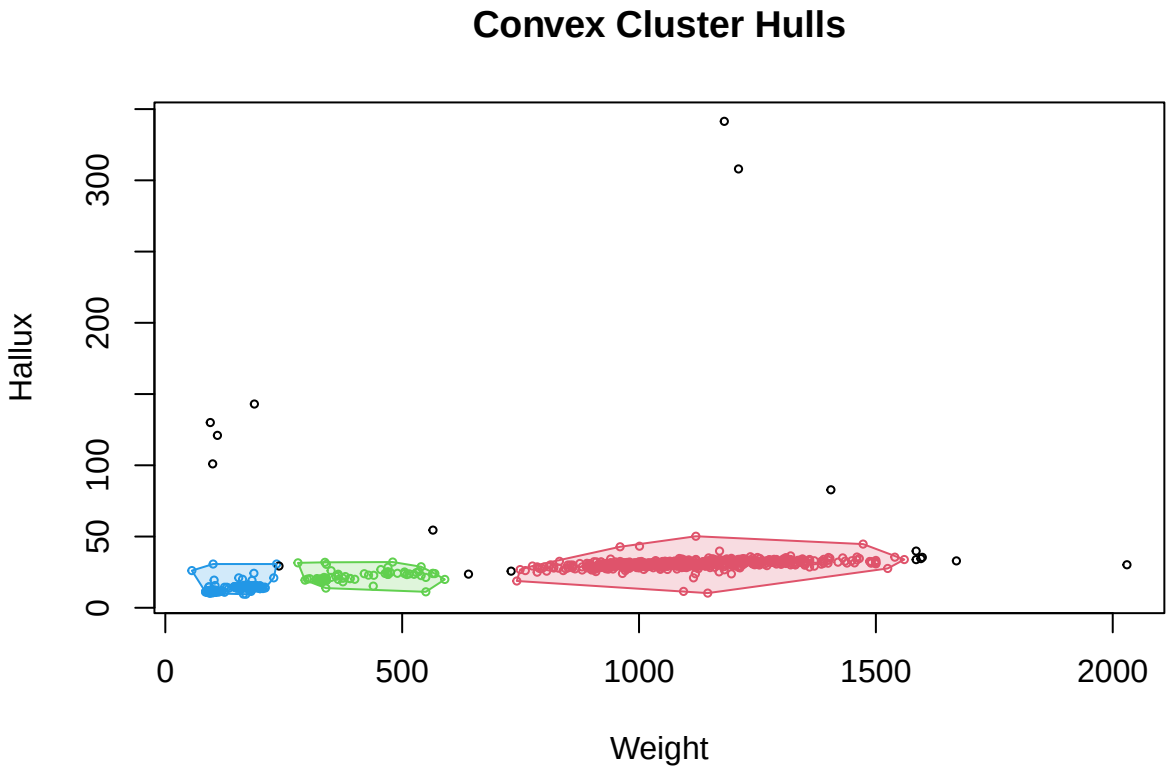
```
opctics_hallux_weight_dbscan <- extractDBSCAN(opctics_hallux_weight, eps_cl = 40)
opctics_hallux_weight_dbscan
```

```
## OPTICS ordering/clustering for 892 objects.
## Parameters: minPts = 10, eps = 530.001141508205, eps_cl = 40, xi = NA
## The clustering contains 3 cluster(s) and 17 noise points.
##
##   0   1   2   3
##  17 555  65 255
##
## Available fields: order, reachdist, coredist, predecessor, minPts, eps,
##                   eps_cl, xi, cluster
```

```
plot(opctics_hallux_weight_dbscan)
```



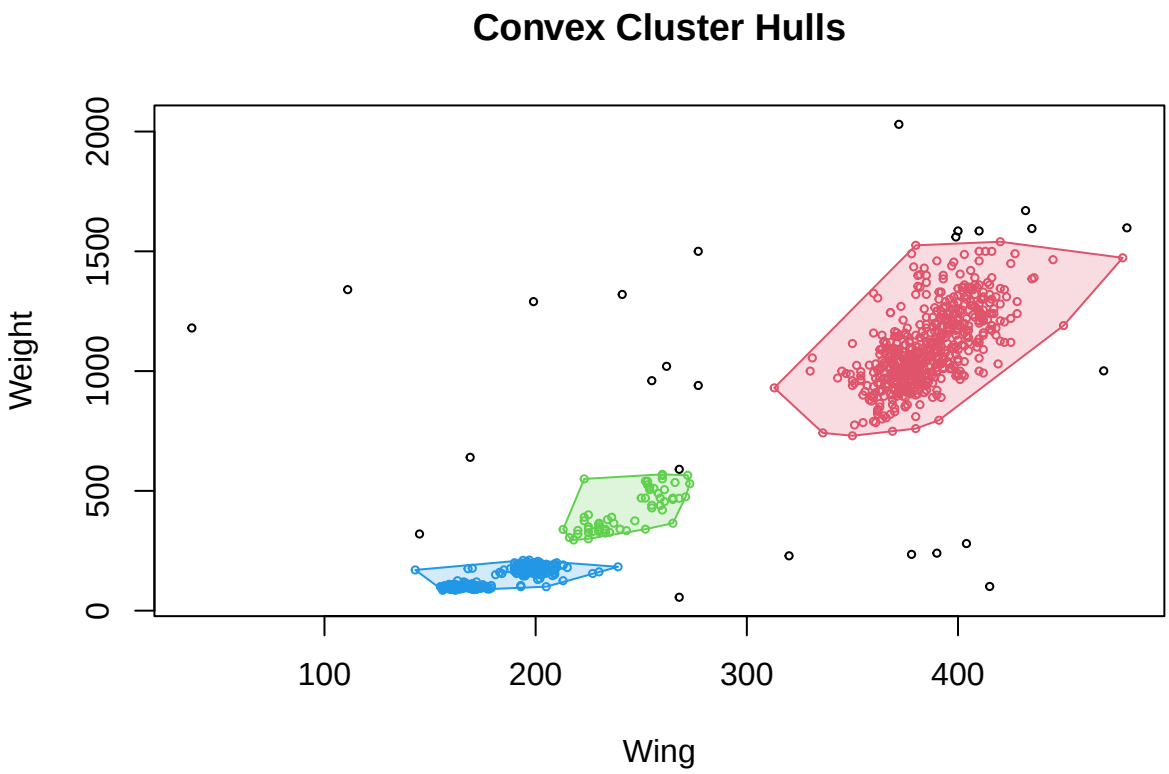

```
hullplot(hawks_k_means_dbscan_hallux_weight, opctics_hallux_weight_dbscan)
```




```
opctics_wing_weight_dbscan <- extractDBSCAN(opctics_wing_weight, eps_cl = 50)
opctics_wing_weight_dbscan
```

```
## OPTICS ordering/clustering for 897 objects.
## Parameters: minPts = 10, eps = 531.583483565846, eps_cl = 50, xi = NA
## The clustering contains 3 cluster(s) and 25 noise points.
##
##   0   1   2   3
##  25 554  62 256
##
## Available fields: order, reachdist, coredist, predecessor, minPts, eps,
##                   eps_cl, xi, cluster
```

```
hullplot(hawks_k_means_dbscan_wing_weight, opctics_wing_weight_dbscan)
```


Convex Cluster Hulls
Weight
0
500
1000
1500
2000
100
200
300
400
Wing