Ejercicio Cluster

Daniel Robins - Fernando Palacios

23 de mayo de 2018

R Markdown

Note that the `echo = FALSE` parameter was added to the code chunk to prevent printing of the R code that generated the plot.

Dataset

El dataset utilizado corresponde a datos de ausentismo en el trabajo obtenido del siguiente sitio https://archive.ics.uci.edu/ml/datasets/Absenteeism+at+work

```
ausentismo <- read.csv("https://raw.githubusercontent.com/fpalaciosdrobins/diplodatos/master/An-lisis%2

summary(ausentismo)
```

```
##        ID        Reason.for.absence Month.of.absence Day.of.the.week
##  Min.   : 1.00   Min.   : 0.00      Min.   : 0.000   Min.   :2.000
##  1st Qu.: 9.00   1st Qu.:13.00      1st Qu.: 3.000   1st Qu.:3.000
##  Median :18.00   Median :23.00      Median : 6.000   Median :4.000
##  Mean   :18.02   Mean   :19.22      Mean   : 6.324   Mean   :3.915
##  3rd Qu.:28.00   3rd Qu.:26.00      3rd Qu.: 9.000   3rd Qu.:5.000
##  Max.   :36.00   Max.   :28.00      Max.   :12.000   Max.   :6.000
##     Seasons      Transportation.expense Distance.from.Residence.to.Work
##  Min.   :1.000   Min.   :118.0          Min.   : 5.00
##  1st Qu.:2.000   1st Qu.:179.0          1st Qu.:16.00
##  Median :3.000   Median :225.0          Median :26.00
##  Mean   :2.545   Mean   :221.3          Mean   :29.63
##  3rd Qu.:4.000   3rd Qu.:260.0          3rd Qu.:50.00
##  Max.   :4.000   Max.   :388.0          Max.   :52.00
##   Service.time        Age        Work.load.Average.day   Hit.target
##  Min.   : 1.00   Min.   :27.00   Min.   :205.9         Min.   : 81.00
##  1st Qu.: 9.00   1st Qu.:31.00   1st Qu.:244.4         1st Qu.: 93.00
##  Median :13.00   Median :37.00   Median :264.2         Median : 95.00
##  Mean   :12.55   Mean   :36.45   Mean   :271.5         Mean   : 94.59
##  3rd Qu.:16.00   3rd Qu.:40.00   3rd Qu.:294.2         3rd Qu.: 97.00
##  Max.   :29.00   Max.   :58.00   Max.   :378.9         Max.   :100.00
##  Disciplinary.failure   Education          Son        Social.drinker
##  Min.   :0.00000      Min.   :1.000   Min.   :0.000   Min.   :0.0000
##  1st Qu.:0.00000      1st Qu.:1.000   1st Qu.:0.000   1st Qu.:0.0000
##  Median :0.00000      Median :1.000   Median :1.000   Median :1.0000
##  Mean   :0.05405      Mean   :1.292   Mean   :1.019   Mean   :0.5676
##  3rd Qu.:0.00000      3rd Qu.:1.000   3rd Qu.:2.000   3rd Qu.:1.0000
##  Max.   :1.00000      Max.   :4.000   Max.   :4.000   Max.   :1.0000
##  Social.smoker          Pet             Weight          Height
##  Min.   :0.00000   Min.   :0.0000   Min.   : 56.00   Min.   :163.0
##  1st Qu.:0.00000   1st Qu.:0.0000   1st Qu.: 69.00   1st Qu.:169.0
##  Median :0.00000   Median :0.0000   Median : 83.00   Median :170.0
##  Mean   :0.07297   Mean   :0.7459   Mean   : 79.04   Mean   :172.1
```

```
##  3rd Qu.:0.00000   3rd Qu.:1.0000   3rd Qu.: 89.00   3rd Qu.:172.0
##  Max.   :1.00000   Max.   :8.0000   Max.   :108.00   Max.   :196.0
##  Body.mass.index Absenteeism.time.in.hours
##  Min.   :19.00   Min.   :  0.000
##  1st Qu.:24.00   1st Qu.:  2.000
##  Median :25.00   Median :  3.000
##  Mean   :26.68   Mean   :  6.924
##  3rd Qu.:31.00   3rd Qu.:  8.000
##  Max.   :38.00   Max.   :120.000
```

Análisis de Variables

Queremos analizar del set de datos y las variables que tenemos continuas son las siguientes:

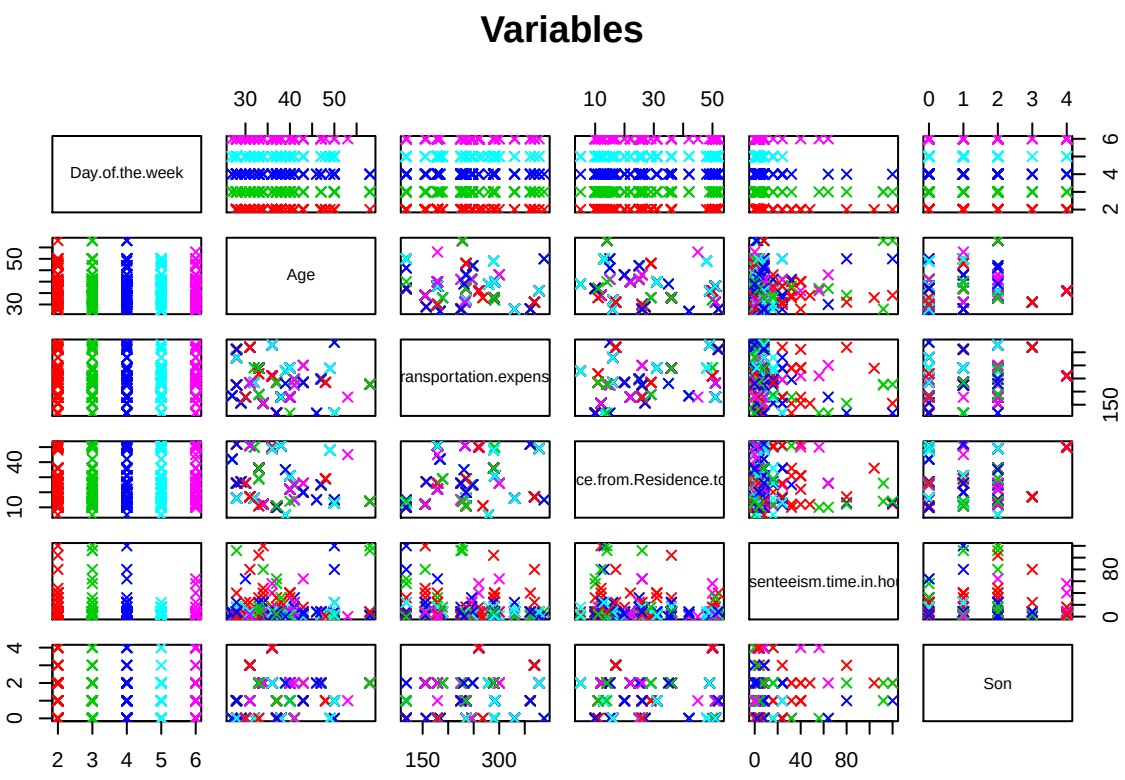
Día de la semana, precio del transporte, distancia desde la casa al trabajo, tiempo de ausentismo en horas y cantidad de hijos

```
#ausentismo$Day.of.the.week
#ausentismo$Age
#ausentismo$Transportation.expense
#ausentismo$Distance.from.Residence.to.Work
#ausentismo$Absenteeism.time.in.hours

analisis <- subset(ausentismo, select=c("Day.of.the.week","Age","Transportation.expense","Distance.from

variables_analizadas = 6

plot(analisis[,1:6], col = analisis$Day.of.the.week, pch = 4, main = "Variables")
```



Aplicamos la función de normalización scale para las variables

```r
# Aplicamos la función de normalización scale para las variables

ausentismo.scale <- scale(analisis[,1:6])

set.seed(100)
ausentismo.km <- kmeans(ausentismo.scale,centers=6)

names(ausentismo.km)
```

```
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```r
print(".....Cluster.....")
```

```
## [1] ".....Cluster....."
```

```r
ausentismo.km$cluster
```

```
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   2   5   6   3   2   6   2   2   5   3   2   2   2   6   6   3   6   6
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   1   3   1   6   2   2   2   2   2   1   2   6   6   3   3   3   6   2
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   2   6   3   2   1   5   6   3   5   5   6   3   2   3   2   1   1   5
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   2   5   1   6   2   3   6   2   6   5   5   1   6   1   5   1   3   3
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   6   1   2   3   1   2   2   2   6   1   6   3   3   5   1   3   1   5
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   1   3   1   2   3   1   2   3   3   4   5   3   1   2   4   2   1   2
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   1   2   5   3   1   1   5   5   5   1   1   1   1   5   1   5   5   5
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   5   5   5   1   2   5   1   5   5   1   2   1   3   6   5   1   2   2
## 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
##   3   3   1   1   1   1   1   1   1   1   1   6   2   1   5   3   2   1
## 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   3   2   5   5   2   5   2   3   2   1   3   6   1   1   1   1   1   6
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
##   3   1   3   1   3   1   5   3   5   1   2   6   5   3   3   3   3   2
## 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216
##   2   5   2   2   3   3   2   5   5   6   1   1   5   6   2   3   1   2
## 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234
##   3   6   5   2   3   2   3   3   3   6   5   3   5   3   5   4   2   6
## 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252
##   3   1   5   5   5   5   1   1   5   1   1   1   2   2   6   2   2   2
## 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270
##   2   1   2   3   3   3   2   2   3   1   1   5   3   3   5   3   2   1
## 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288
##   6   6   2   3   3   3   3   6   6   3   6   3   2   3   5   3   3   3
## 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306
##   6   2   3   2   2   5   3   1   5   2   6   4   3   2   2   3   3   3
## 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324
##   5   3   1   5   2   2   3   1   1   1   5   5   3   6   2   3   2   4
```

```
## 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342
##   3   1   3   5   3   6   3   3   3   6   2   2   2   3   6   1   6   6
## 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360
##   3   5   2   3   6   6   1   5   3   6   3   3   2   1   6   2   1   5
## 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378
##   5   3   5   2   3   3   5   6   6   5   6   1   2   2   6   6   6   6
## 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396
##   3   6   1   6   3   3   1   6   6   3   6   6   5   3   3   2   6   3
## 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414
##   5   2   6   2   2   3   5   5   5   3   3   5   6   2   2   3   6   2
## 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432
##   1   2   3   1   6   5   4   3   2   6   5   3   3   2   1   6   2   2
## 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450
##   5   5   3   1   1   1   5   3   1   5   6   5   1   2   6   3   6   3
## 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468
##   6   3   6   1   5   6   1   3   1   3   1   1   2   1   1   5   2   5
## 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486
##   3   3   3   4   6   5   2   2   3   3   3   3   6   5   3   5   2   3
## 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504
##   3   6   5   1   3   3   5   3   2   3   3   5   5   3   1   3   6   1
## 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522
##   1   2   2   3   1   3   3   3   5   3   3   1   2   2   1   2   6   3
## 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540
##   3   2   1   2   1   1   1   3   1   1   6   5   1   6   5   2   1   1
## 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558
##   2   3   5   1   6   5   5   1   3   2   3   3   1   2   2   6   3   1
## 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576
##   1   1   1   2   3   3   6   1   1   1   1   4   1   1   3   3   1   3
## 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594
##   3   3   1   5   1   6   2   6   6   6   1   3   6   1   6   1   3   6
## 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612
##   6   6   1   6   6   1   1   3   6   6   1   6   6   6   5   1   6   1
## 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630
##   6   3   1   6   6   1   6   3   3   3   4   6   1   6   1   1   1   2
## 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648
##   6   6   6   1   6   3   6   1   6   2   3   6   3   6   1   6   6   5
## 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666
##   5   6   6   1   4   2   1   5   2   3   1   1   2   5   5   2   5   3
## 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684
##   1   1   1   1   5   1   5   1   3   3   1   5   3   1   3   3   4   3
## 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702
##   1   5   5   5   5   1   2   3   4   1   3   3   1   1   1   3   1   2
## 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720
##   5   3   5   1   5   1   5   5   2   5   6   1   3   1   5   1   3   1
## 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738
##   5   6   5   5   6   3   6   3   5   4   1   5   2   1   4   2   3   5
## 739 740
##   3   3
```

```
print(".....Centers.....")
```

```
## [1] ".....Centers....."
```

```
ausentismo.km$centers
```

```
##   Day.of.the.week        Age Transportation.expense
## 1     0.047615164 -1.1382432            -0.05042762
## 2     0.137423388 -0.5300776             1.28615916
## 3     0.079207755  0.9194015             0.46260552
## 4    -0.543026978  0.5148683            -0.36659000
## 5    -0.274517580  0.5694511            -1.36581699
## 6     0.008415711  0.1551659            -0.54553605
##   Distance.from.Residence.to.Work Absenteeism.time.in.hours         Son
## 1                      -0.3082164                -0.2010456 -0.44063727
## 2                       0.8591395                 0.0600169  1.42355383
## 3                      -0.4247153                -0.1006606  0.08281484
## 4                      -0.8368924                 6.0388762  0.37292363
## 5                      -1.1806010                -0.1191403 -0.14407352
## 6                       1.4386219                -0.2010669 -0.87575600
```

```
print(".........totss.............")
```

```
## [1] ".........totss............."
```

```
ausentismo.km$totss
```

```
## [1] 4434
```

```
print(".........withinss.........")
```

```
## [1] ".........withinss........."
```

```
ausentismo.km$withinss
```

```
## [1] 331.2842 483.9045 507.5946 122.3459 335.6493 167.1495
```

```
print(".........tot.withinss......")
```

```
## [1] ".........tot.withinss......"
```

```
ausentismo.km$tot.withinss
```

```
## [1] 1947.928
```

```
print(".........betweenss.........")
```

```
## [1] ".........betweenss........."
```

```
ausentismo.km$betweenss
```

```
## [1] 2486.072
```

```
print(".........size.............")
```

```
## [1] ".........size............."
```

```
ausentismo.km$size
```

```
## [1] 172 127 182  14 122 123
```

```
print(".........iter.............")
```

```
## [1] ".........iter............."
```

```
ausentismo.km$iter
```

```
## [1] 4
```

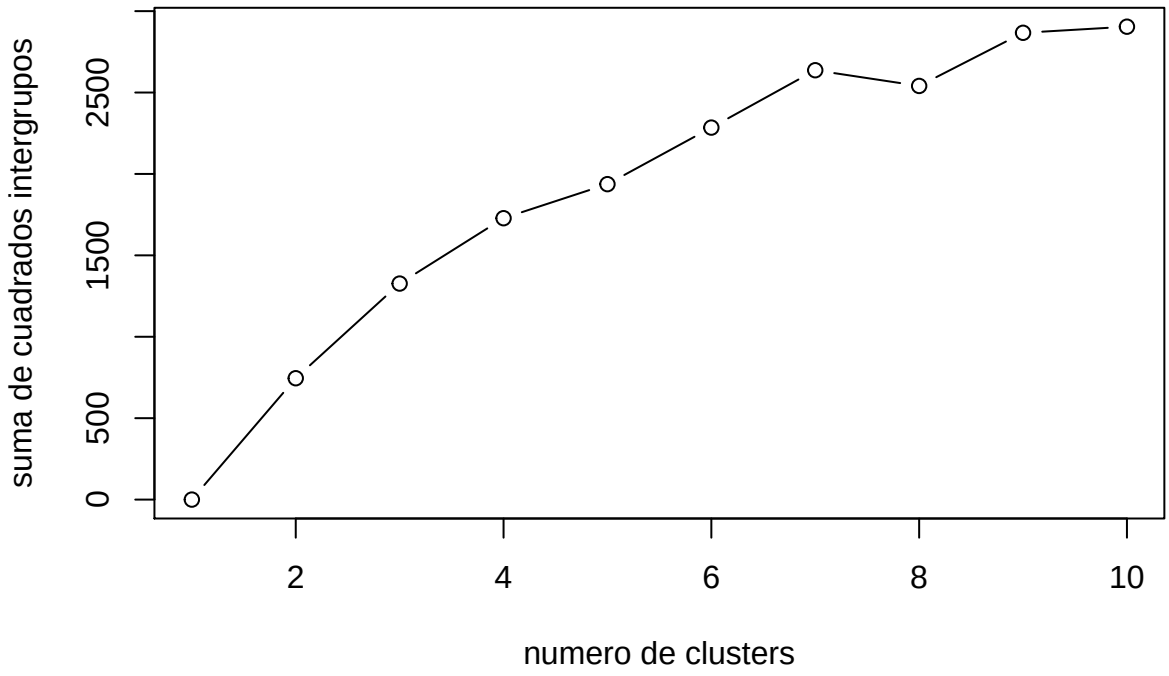
Determinar el número óptimo de Clusters por la suma de distancias interclusters

```
# Determinar el número óptimo de Clusters

sumbt <- kmeans(ausentismo.scale,centers = 1)$betweens

for (i in 2:10) sumbt[i] <- kmeans(ausentismo.scale,centers = i)$betweenss

plot(1:10,sumbt,type="b",xlab="numero de clusters",ylab = "suma de cuadrados intergrupos")
```

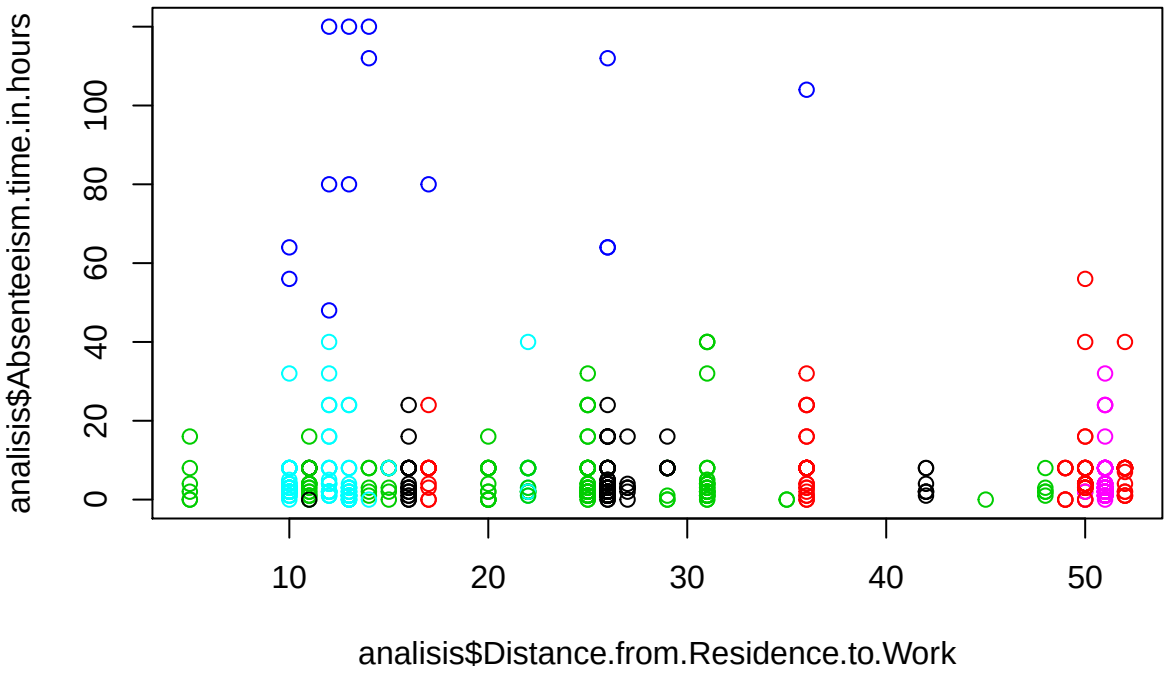


#

Análisis de Resultados

```
# Inspeccionando los resultados

plot(analisis$Distance.from.Residence.to.Work,analisis$Absenteeism.time.in.hours,col=ausentismo.km$clus
```




```
aggregate(analisis[,1:6], by=list(ausentismo.km$cluster),median)
```

```
##   Group.1 Day.of.the.week  Age Transportation.expense
## 1       1               4 28.0                   225
## 2       2               4 33.0                   289
## 3       3               4 41.0                   246
## 4       4               3 35.5                   167
## 5       5               3 37.0                   118
## 6       6               4 38.0                   179
##   Distance.from.Residence.to.Work Absenteeism.time.in.hours Son
## 1                            26.0                         3   0
## 2                            50.0                         8   2
## 3                            25.0                         3   1
## 4                            13.5                        80   2
## 5                            12.0                         3   1
## 6                            51.0                         3   0
```

Análisis por otros métodos de Clustering

Normalización

```
normalize <- function(x) {
  return ((x-min(x))/(max(x)-min(x)))
}
```

```
analisis_n <- as.data.frame(lapply(analisis[2:variables_analizadas], normalize))
```

```
analisis_train <- analisis_n[1:222, ]
analisis_test  <- analisis_n[223:740, ]

analisis_train_labels <- analisis[1:222, 1]
```

```r
analisis_test_labels  <- analisis[223:740, 1]
```

```r
library(class)
analisis_test_pred <- knn(train=analisis_train, test=analisis_test, cl=analisis_train_labels, k=20)
library(gmodels)
CrossTable(x=analisis_test_labels, y=analisis_test_pred, prop.chisq = FALSE)
```

```
##
##
##    Cell Contents
## |-------------------------|
## |                       N |
## |           N / Row Total |
## |           N / Col Total |
## |         N / Table Total |
## |-------------------------|
##
##
## Total Observations in Table:  518
##
##
##                      | analisis_test_pred
## analisis_test_labels |         2 |         3 |         4 |         5 |         6 | Row Total |
## ---------------------|-----------|-----------|-----------|-----------|-----------|-----------|
##                    2 |         2 |        38 |        38 |        18 |        25 |       121 |
##                      |     0.017 |     0.314 |     0.314 |     0.149 |     0.207 |     0.234 |
##                      |     0.400 |     0.215 |     0.248 |     0.290 |     0.207 |           |
##                      |     0.004 |     0.073 |     0.073 |     0.035 |     0.048 |           |
## ---------------------|-----------|-----------|-----------|-----------|-----------|-----------|
##                    3 |         3 |        34 |        26 |        19 |        23 |       105 |
##                      |     0.029 |     0.324 |     0.248 |     0.181 |     0.219 |     0.203 |
##                      |     0.600 |     0.192 |     0.170 |     0.306 |     0.190 |           |
##                      |     0.006 |     0.066 |     0.050 |     0.037 |     0.044 |           |
## ---------------------|-----------|-----------|-----------|-----------|-----------|-----------|
##                    4 |         0 |        42 |        36 |         5 |        20 |       103 |
##                      |     0.000 |     0.408 |     0.350 |     0.049 |     0.194 |     0.199 |
##                      |     0.000 |     0.237 |     0.235 |     0.081 |     0.165 |           |
##                      |     0.000 |     0.081 |     0.069 |     0.010 |     0.039 |           |
## ---------------------|-----------|-----------|-----------|-----------|-----------|-----------|
##                    5 |         0 |        29 |        29 |        11 |        21 |        90 |
##                      |     0.000 |     0.322 |     0.322 |     0.122 |     0.233 |     0.174 |
##                      |     0.000 |     0.164 |     0.190 |     0.177 |     0.174 |           |
##                      |     0.000 |     0.056 |     0.056 |     0.021 |     0.041 |           |
## ---------------------|-----------|-----------|-----------|-----------|-----------|-----------|
##                    6 |         0 |        34 |        24 |         9 |        32 |        99 |
##                      |     0.000 |     0.343 |     0.242 |     0.091 |     0.323 |     0.191 |
##                      |     0.000 |     0.192 |     0.157 |     0.145 |     0.264 |           |
##                      |     0.000 |     0.066 |     0.046 |     0.017 |     0.062 |           |
## ---------------------|-----------|-----------|-----------|-----------|-----------|-----------|
##         Column Total |         5 |       177 |       153 |        62 |       121 |       518 |
##                      |     0.010 |     0.342 |     0.295 |     0.120 |     0.234 |           |
## ---------------------|-----------|-----------|-----------|-----------|-----------|-----------|
##
##
```

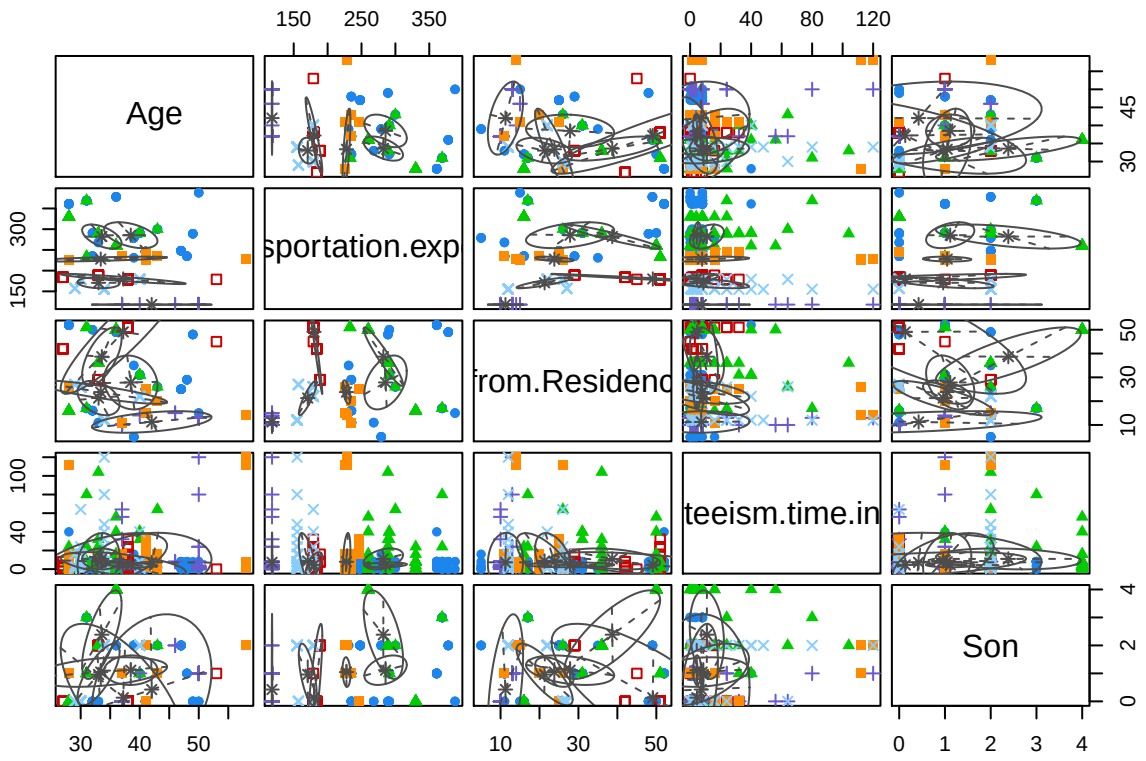
```r
# Mclust comes with a method of hierarchical clustering.
library(mclust)
```

```
## Package 'mclust' version 5.4
## Type 'citation("mclust")' for citing this R package in publications.
```

```r
# initialize 3 different classes.
analisis_initialk <- mclust::hc(data = analisis, modelName = "EII")
analisis_initialk <- mclust::hclass(analisis_initialk, 6)
```

```r
# Select 4 continuous variables and look for dos distinct groups.
analisis_mcl.model <- Mclust(analisis[, 2:variables_analizadas], 6)
# Plot our results.

plot(analisis_mcl.model, what = "classification", main = "Mclust Classification")
```




```r
summary(analisis_mcl.model)
```

```
## ----------------------------------------------------
## Gaussian finite mixture model fitted by EM algorithm
## ----------------------------------------------------
##
## Mclust EEV (ellipsoidal, equal volume and shape) model with 6 components:
##
##  log.likelihood   n  df       BIC       ICL
##       -11765.84 740 100 -24192.34 -24205.73
##
## Clustering table:
##   1   2   3   4   5   6
## 193 129 112  92 112 102
```

```r
### ejemplo de K-means:

set.seed(20)
analisisCluster <- kmeans(analisis[, 2:variables_analizadas], 6, nstart = 20)
analisisCluster
```

```
## K-means clustering with 6 clusters of sizes 66, 117, 207, 8, 201, 141
##
## Cluster means:
##        Age Transportation.expense Distance.from.Residence.to.Work
## 1 30.65152               358.5909                         33.27273
## 2 40.04274               126.5385                         11.40171
## 3 36.56522               233.9275                         23.49758
## 4 42.75000               175.7500                         16.25000
## 5 35.58706               178.8060                         40.46269
## 6 36.88652               280.4468                         37.37589
##   Absenteeism.time.in.hours       Son
## 1                  8.227273 1.2878788
## 2                  6.384615 0.7777778
## 3                  4.536232 0.9806763
## 4                101.000000 1.3750000
## 5                  4.611940 0.2835821
## 6                  8.226950 2.1773050
##
## Clustering vector:
##   1   2   3   4   5   6   7   8   9  10  11  12  13  14  15  16  17  18
##   6   2   5   6   6   5   1   6   2   3   6   6   6   5   5   3   5   5
##  19  20  21  22  23  24  25  26  27  28  29  30  31  32  33  34  35  36
##   5   3   1   5   1   6   6   1   6   5   6   5   5   3   3   3   5   1
##  37  38  39  40  41  42  43  44  45  46  47  48  49  50  51  52  53  54
##   6   6   3   6   5   2   5   3   2   2   5   6   6   6   6   3   3   2
##  55  56  57  58  59  60  61  62  63  64  65  66  67  68  69  70  71  72
##   6   2   3   5   1   3   5   6   5   2   2   5   5   3   2   3   3   6
##  73  74  75  76  77  78  79  80  81  82  83  84  85  86  87  88  89  90
##   5   3   6   6   3   6   1   6   5   3   5   5   6   2   5   6   3   2
##  91  92  93  94  95  96  97  98  99 100 101 102 103 104 105 106 107 108
##   3   5   3   6   3   3   6   6   3   2   2   3   3   6   2   1   3   6
## 109 110 111 112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
##   3   1   2   3   3   3   2   2   2   3   5   3   3   2   3   2   2   2
## 127 128 129 130 131 132 133 134 135 136 137 138 139 140 141 142 143 144
##   2   2   2   5   6   2   5   2   2   3   6   5   3   5   2   5   6   6
## 145 146 147 148 149 150 151 152 153 154 155 156 157 158 159 160 161 162
##   1   1   3   3   5   5   3   1   1   3   5   6   6   5   5   6   6   5
## 163 164 165 166 167 168 169 170 171 172 173 174 175 176 177 178 179 180
##   3   6   5   2   6   2   6   5   6   3   6   5   3   3   5   3   3   5
## 181 182 183 184 185 186 187 188 189 190 191 192 193 194 195 196 197 198
##   6   3   3   3   6   3   2   3   2   3   6   5   2   6   3   6   3   6
## 199 200 201 202 203 204 205 206 207 208 209 210 211 212 213 214 215 216
##   6   2   1   6   3   3   6   2   2   5   3   5   2   5   6   6   1   1
## 217 218 219 220 221 222 223 224 225 226 227 228 229 230 231 232 233 234
##   1   5   2   1   3   1   3   6   3   5   2   3   2   3   2   4   1   5
## 235 236 237 238 239 240 241 242 243 244 245 246 247 248 249 250 251 252
##   6   3   2   2   2   2   1   5   2   1   1   5   1   6   5   6   6   6
## 253 254 255 256 257 258 259 260 261 262 263 264 265 266 267 268 269 270
```

```
##   6   5   6   3   6   6   6   6   3   6   5   2   3   3   2   3   1   5
## 271 272 273 274 275 276 277 278 279 280 281 282 283 284 285 286 287 288
##   5   5   6   3   3   6   3   6   5   3   5   6   6   3   2   3   6   6
## 289 290 291 292 293 294 295 296 297 298 299 300 301 302 303 304 305 306
##   5   6   3   6   6   2   3   5   2   1   5   4   3   6   6   3   3   3
## 307 308 309 310 311 312 313 314 315 316 317 318 319 320 321 322 323 324
##   2   6   5   2   1   6   6   5   5   5   2   2   3   5   6   6   1   4
## 325 326 327 328 329 330 331 332 333 334 335 336 337 338 339 340 341 342
##   3   1   3   2   3   5   3   6   3   5   6   6   1   6   5   3   5   5
## 343 344 345 346 347 348 349 350 351 352 353 354 355 356 357 358 359 360
##   3   2   6   3   5   5   5   2   3   5   3   3   6   5   5   6   3   2
## 361 362 363 364 365 366 367 368 369 370 371 372 373 374 375 376 377 378
##   2   3   2   6   6   3   2   5   5   2   5   3   6   6   5   5   5   5
## 379 380 381 382 383 384 385 386 387 388 389 390 391 392 393 394 395 396
##   3   5   5   5   3   6   5   5   5   3   5   5   2   3   3   6   5   3
## 397 398 399 400 401 402 403 404 405 406 407 408 409 410 411 412 413 414
##   5   1   5   1   1   3   2   2   2   3   3   2   5   6   6   3   5   6
## 415 416 417 418 419 420 421 422 423 424 425 426 427 428 429 430 431 432
##   1   1   3   1   5   2   4   6   6   5   2   3   3   1   5   5   1   6
## 433 434 435 436 437 438 439 440 441 442 443 444 445 446 447 448 449 450
##   5   5   3   3   1   5   2   3   5   2   5   2   3   6   5   6   5   3
## 451 452 453 454 455 456 457 458 459 460 461 462 463 464 465 466 467 468
##   5   3   5   3   2   5   5   3   1   3   5   3   6   6   1   2   6   2
## 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484 485 486
##   6   3   3   4   5   2   1   6   3   3   6   6   5   5   3   2   6   3
## 487 488 489 490 491 492 493 494 495 496 497 498 499 500 501 502 503 504
##   6   5   5   1   3   3   2   6   6   3   3   2   2   6   5   3   5   3
## 505 506 507 508 509 510 511 512 513 514 515 516 517 518 519 520 521 522
##   5   1   1   6   3   3   6   6   2   6   6   3   1   1   3   1   5   3
## 523 524 525 526 527 528 529 530 531 532 533 534 535 536 537 538 539 540
##   6   1   3   1   3   5   3   3   3   3   5   2   1   5   2   6   3   3
## 541 542 543 544 545 546 547 548 549 550 551 552 553 554 555 556 557 558
##   1   6   2   3   5   2   2   3   6   6   3   3   3   1   1   5   6   3
## 559 560 561 562 563 564 565 566 567 568 569 570 571 572 573 574 575 576
##   5   3   3   1   5   3   3   5   3   3   3   4   5   5   5   5   5   5
## 577 578 579 580 581 582 583 584 585 586 587 588 589 590 591 592 593 594
##   6   5   5   2   5   5   6   5   5   5   3   3   5   3   5   5   3   5
## 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612
##   3   5   3   5   5   3   5   5   5   3   5   5   5   5   2   3   5   3
## 613 614 615 616 617 618 619 620 621 622 623 624 625 626 627 628 629 630
##   5   3   3   5   5   3   5   3   3   3   4   5   3   5   3   5   3   1
## 631 632 633 634 635 636 637 638 639 640 641 642 643 644 645 646 647 648
##   5   5   5   5   5   3   5   3   5   6   3   5   3   5   5   5   5   2
## 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666
##   2   5   5   5   6   1   3   2   1   3   5   5   1   5   2   1   2   3
## 667 668 669 670 671 672 673 674 675 676 677 678 679 680 681 682 683 684
##   5   3   3   5   2   3   2   5   3   3   3   2   3   5   6   3   6   3
## 685 686 687 688 689 690 691 692 693 694 695 696 697 698 699 700 701 702
##   5   2   2   2   2   5   1   5   2   3   6   5   3   1   3   6   5   1
## 703 704 705 706 707 708 709 710 711 712 713 714 715 716 717 718 719 720
##   2   5   2   3   2   5   2   2   1   2   3   5   3   6   2   5   6   5
## 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738
##   2   3   2   2   3   3   3   3   2   4   5   2   1   3   1   6   3   2
## 739 740
```

```
##   3   5
##
## Within cluster sum of squares by cluster:
## [1] 49995.15 46587.90 45327.93 20368.38 44926.70 71883.42
##  (between_SS / total_SS =  92.3 %)
##
## Available components:
##
## [1] "cluster"      "centers"      "totss"        "withinss"
## [5] "tot.withinss" "betweenss"    "size"         "iter"
## [9] "ifault"
```

```
table(analisisCluster$cluster, analisis$Day.of.the.week)
```

```
##
##      2  3  4  5  6
##   1 17 17 15  7 10
##   2 27 38 23 16 13
##   3 46 44 45 34 38
##   4  1  4  3  0  0
##   5 45 30 45 38 43
##   6 25 21 25 30 40
```

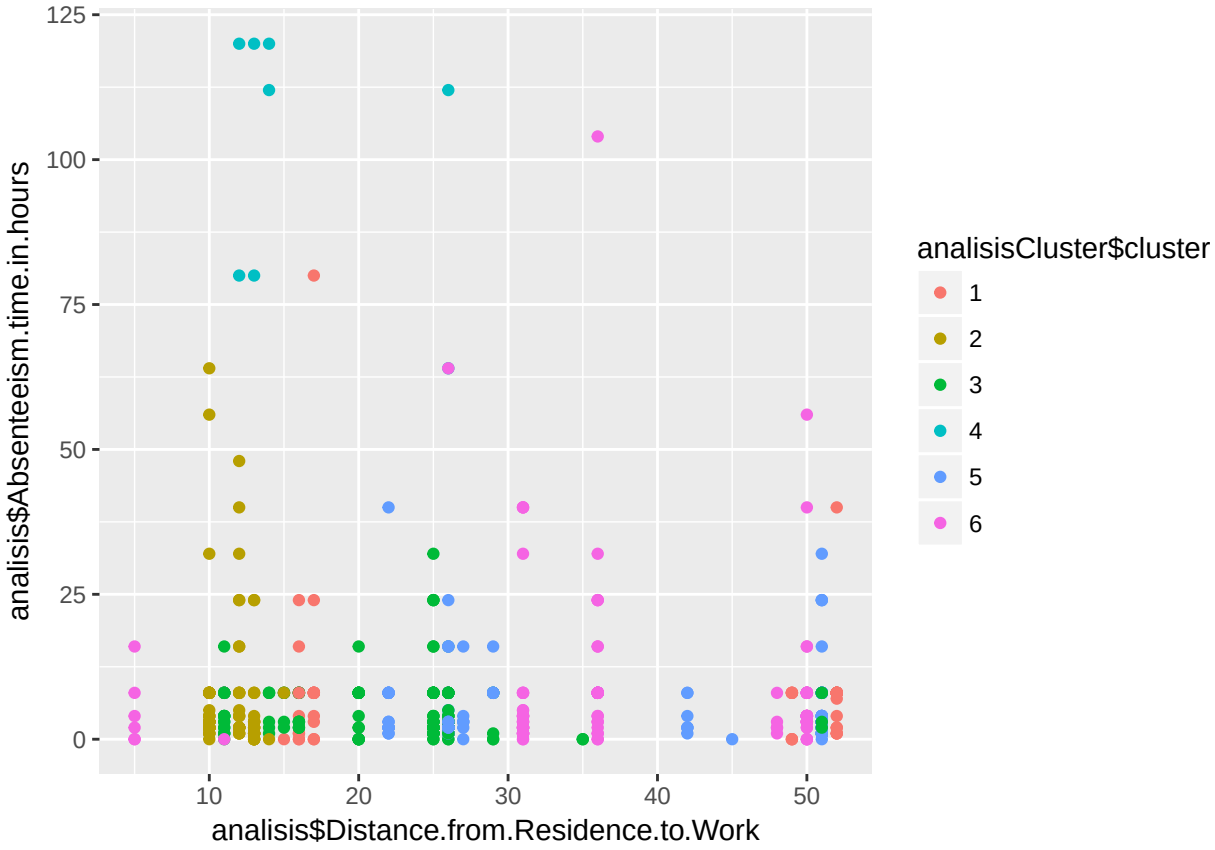
```
aggregate(analisis[,1:variables_analizadas], by=list(analisisCluster$cluster),median)
```

```
##   Group.1 Day.of.the.week Age Transportation.expense
## 1       1               3  28                    361
## 2       2               3  37                    118
## 3       3               4  37                    235
## 4       4               3  42                    167
## 5       5               4  38                    179
## 6       6               4  36                    289
##   Distance.from.Residence.to.Work Absenteeism.time.in.hours Son
## 1                            17.0                        8 1.0
## 2                            12.0                        3 1.0
## 3                            25.0                        3 1.0
## 4                            13.5                      112 1.5
## 5                            51.0                        3 0.0
## 6                            36.0                        4 2.0
```

```
plot(cbind(analisis$Distance.from.Residence.to.Work, analisis$Absenteeism.time.in.hours),col=analisisClu

points(analisisCluster$centers[,3:4],col=11,pch=19,cex=3)
points(matrix(colMeans(analisisCluster$centers[,3:4]),nrow = 1,ncol = 2),col=12,pch=19,cex=3)
```


```
library(ggplot2)
analisisCluster$cluster <- as.factor(analisisCluster$cluster)
ggplot(analisis, aes(analisis$Distance.from.Residence.to.Work, analisis$Absenteeism.time.in.hours, colo
```




```
tabla_analisis_gastos <- table(analisisCluster$cluster, analisis$Transportation.expense)
tabla_analisis_horas <- table(analisisCluster$cluster, analisis$Absenteeism.time.in.hours)
tabla_analisis_gastos
```

```
##
##     118 155 157 179 184 189 225 228 231 233 235 246 248 260 268 279 289
##   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   2  90  27   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0
##   3   0   0   0   0   0   0  80   6   2   7  58  30  24   0   0   0   0
##   4   2   2   0   1   0   0   1   2   0   0   0   0   0   0   0   0   0
##   5   0   0   7 179   7   8   0   0   0   0   0   0   0   0   0   0   0
##   6   0   0   0   0   0   0   0   0   0   0   0   0   0  42   3   6  45
##
##     291 300 330 361 369 378 388
##   1   0   0  16  24  15   8   3
##   2   0   0   0   0   0   0   0
##   3   0   0   0   0   0   0   0
##   4   0   0   0   0   0   0   0
##   5   0   0   0   0   0   0   0
##   6  40   5   0   0   0   0   0
```

```
tabla_analisis_horas
```

```
##
##      0  1  2  3  4  5  7  8 16 24 32 40 48 56 64 80 104 112 120
##   1  8  5  2  1  3  0  1 41  1  2  0  1  0  0  0  1   0   0   0
##   2  8 17 32 16  8  2  0 22  2  4  2  1  1  1  1  0   0   0   0
##   3 14 26 50 36 13  1  0 58  5  3  1  0  0  0  0  0   0   0   0
```

```
##   4  0  0  0  0  0  0  0  0  0  0  0  0  0  0  1  2  0  2  3
##   5  3 29 58 44 12  2  0 41  6  4  1  1  0  0  0  0  0  0  0
##   6 11 11 15 15 24  2  0 46  5  3  2  4  0  1  1  0  1  0  0
```

- Investigue los resultados en el meta parametro K numero de cumulos e investigue posibles procesos de seleccion del mismo.

La cantidad de cúmulos que nos permitió encontrar un grupo de datos interesantes son 6, y el procesos de selección que utilizamos es la suma de las distancias interclusters (between) graficadas, buscando donde las sumas acumuladas muestran decrecimiento en el gráfico

Una vez graficado, hicimos iteraciones con diferentes cúmulos entre 4 y 7 hasta encontrar datos que nos parecian representativos

- Elabore un resumen, y selecione un mejor valor segun el/los criterios aplicados, discuta el significado de los cumulos encontrados.

Hemos aplicado varios criterios de clustering, en primer lugar trabajamos con un dataset de una encuesta con variables discretas, cuyos valores estaban casi todas entre 1 y 5 y no nos permitió encontrar datos que nos sirvieran, entendemos que al ser todas las variables similares, las distancias para generar los clusters son también similares, por lo cual es mejor aplicar estos métodos con variables contínuas.

En este sentido cambiamos el dataset e hicimos el análisis que mostramos en este práctico.

Las pruebas que hicimos con mclust como con knn no nos dieron agrupamientos que nos dijeran mucho a cerca de los datos analizados.

El método que si nos permitió encontrar valores interesantes es kmeans

Pudimos distinguir dos grupos de trabajadores 1) Alto grado de ausentismo: (Cluster 4) Representado por un grupo que viven cerca del trabajo, con dos hijos, que faltan el día de la semana 3 y edad moda de 40 años 2) Altos costos de Transporte: (Cluster 1) Otro grupo con altos costos de transporte cuyos trabajadores son jóvenes que viven cerca del trabajo y los costos de transporte son altos, el ausentismo en este caso duplica a otro grupo cuyas distancias son similares pero el costo de transporte es bajo (Cluster 2)

- Comente la influencia de la normalizacion de los datos en los resultados del clustering.

Al tener variables con valores de escala muy diferentes, como pueden ser los dias de ausentismo y la cantidad de hijos o el gasto en transporte, si no normalizamos, la prevalencia de variables con valores más altos influyen en los agrupamientos, por lo cual es necesario normalizar todas las variables. Utilizamos dos métodos de normalización (scale y la función de suma de suma de cuadrados)