

# Mineria de datos

*David Cordoba Ruiz, Laura Lopez Parrilla, M.C. Galvez Ortiz, Victor Valero Fernandez*

## Contents

<b>1 Lectura de librerias principales</b>	<b>1</b>
<b>2 Lectura y descripción de los datos</b>	<b>2</b>
<b>3 Limpieza inicial y selección de datos útiles</b>	<b>19</b>
3.1 Limpieza adicional . . . . .	22
<b>4 Revisión de los datos faltantes</b>	<b>24</b>
<b>5 Exploración de datos: revisión de dependencias entre variables</b>	<b>24</b>
<b>6 Reducción de variables</b>	<b>28</b>
<b>7 Filtrado de Galaxias y Estrellas</b>	<b>29</b>
<b>8 Análisis con clusterización</b>	<b>30</b>
<b>9 Regresión logistica</b>	<b>35</b>
9.1 Ficheros de entrenamiento y testeо: . . . . .	35
9.2 Regresión logística con todas las variables usando el fichero de entrenamiento . . . . .	35
9.3 Regresion logística eliminando las dos variables: . . . . .	37
9.4 Regresión logística para fichero con reducción de variables . . . . .	40
<b>10 Predicción de datos</b>	<b>42</b>
<b>11 Evaluación de modelos. Comparación</b>	<b>45</b>
<b>12 Modelo con Máquinas de Vector Soporte</b>	<b>46</b>
12.1 SVM no lineal . . . . .	47
<b>13 Estudio con Árboles de decisión</b>	<b>50</b>
13.1 Nuevo fichero de datos . . . . .	51
13.2 Selección de datos útiles . . . . .	52
13.3 Limpieza . . . . .	52
13.4 Ficheros de entrenamiento y testeо . . . . .	53
13.5 Regresión logística para estudio de dependencia de variables . . . . .	54
13.6 Árbol de decisión . . . . .	55
<b>14 Resumen y conclusiones</b>	<b>64</b>

\*NOTA: Laura Lopez Parrilla ha participado en la parte correspondiente a Minería de datos I pero no en la parte correspondiente a Minería de datos II.

## 1 Lectura de librerias principales

```
library(tidyr)
library(dplyr)
```

```

library(ggplot2)
library(VIM)
library(Hmisc)
library(corrplot)
library(car)
library(cowplot)
library(data.table)
library(pROC)
library(ROCR)
library(corrplot)

```

## 2 Lectura y descripción de los datos

```

set.seed(31415)
data <- read.csv("catalogoalhambra.csv")
head(data)

##          RA      DEC      dis     objID Field Pointing CCD      x      y
## 1 356.9498 15.3437 56292.46 81481409802     8       1   4 665.911 585.596
## 2 356.9362 15.3427 56298.27 81481409855     8       1   4 879.360 570.421
## 3 356.9523 15.3459 56298.51 81481409782     8       1   4 626.952 621.435
## 4 356.9404 15.3439 56299.62 81481409788     8       1   4 813.644 588.895
## 5 356.9412 15.3442 56300.12 81481409783     8       1   4 800.579 593.241
## 6 356.9370 15.3438 56301.60 81481409801     8       1   4 867.043 587.045
##    area   fwhm  stell   ell     a     b theta   rk   rf s2n photoflag
## 1   41 11.12  0.26 0.3550 2.310 1.490 -48.7 3.50 2.923 11.34      0
## 2    5  5.00  0.40 0.2988 1.004 0.704 -34.6 6.32 2.070  4.09      0
## 3   16  7.14  0.52 0.4132 1.561 0.916  89.0 6.75 3.328  5.84      0
## 4   79 11.77  0.15 0.1965 2.880 2.314  25.8 4.78 4.049 12.04      3
## 5   72  8.01  0.11 0.1350 2.370 2.050 -60.2 3.50 2.930 16.16      0
## 6    8  5.81  0.38 0.4655 1.334 0.713  56.4 3.50 1.794  4.45      2
##    F365W dF365W  F396W dF396W  F427W dF427W  F458W dF458W  F489W dF489W
## 1 25.507  0.675 99.000 26.120 99.000 26.110 25.459  0.740 25.819  0.706
## 2 99.000 25.990 99.000 26.120 99.000 26.110 99.000 25.940 99.000 26.220
## 3 99.000 25.990 99.000 26.120 99.000 26.110 25.282  0.727 99.000 26.220
## 4 24.397  0.345 25.152  0.559 24.910  0.467 24.463  0.425 23.796  0.168
## 5 23.916  0.165 24.096  0.158 23.779  0.123 24.198  0.249 23.977  0.146
## 6 99.000 25.990 99.000 26.120 99.000 26.110 99.000 25.940 26.130  0.580
##    F520W dF520W  F551W dF551W  F582W dF582W  F613W dF613W  F644W dF644W
## 1 25.352  0.566 24.706  0.109 25.057  0.135 25.308  0.173 24.832  0.250
## 2 99.000 25.900 25.353  0.251 25.819  0.331 25.512  0.287 25.635  0.684
## 3 24.820  0.556 25.160  0.203 25.230  0.207 26.039  0.448 25.058  0.336
## 4 24.241  0.279 24.078  0.081 24.441  0.121 24.246  0.105 24.256  0.227
## 5 24.108  0.187 23.799  0.049 23.931  0.057 23.716  0.050 24.004  0.133
## 6 99.000 25.900 25.963  0.205 26.620  0.344 26.092  0.234 26.331  0.438
##    F675W dF675W  F706W dF706W  F737W dF737W  F768W dF768W  F799W dF799W
## 1 24.499  0.185 24.460  0.071 24.526  0.242 24.488  0.372 24.029  0.075
## 2 25.298  0.563 25.473  0.166 26.115  1.188 25.647  1.087 99.000 26.930
## 3 25.425  0.488 25.510  0.183 99.000 26.020 23.827  0.254 24.201  0.118
## 4 23.946  0.169 24.094  0.086 24.115  0.246 23.611  0.253 23.199  0.057
## 5 23.965  0.127 23.571  0.041 24.029  0.168 23.871  0.237 23.586  0.060

```

```

## 6 27.155 0.936 26.809 0.254 99.000 26.020 25.764 0.531 25.545 0.183
## F830W dF830W F861W dF861W F892W dF892W F923W dF923W F954W dF954W
## 1 24.901 0.695 23.695 0.151 23.555 0.270 23.812 0.649 24.002 1.020
## 2 24.820 0.703 25.238 0.531 24.637 0.717 99.000 23.570 99.000 22.910
## 3 24.966 0.818 25.372 0.686 99.000 24.720 24.130 1.068 23.077 0.567
## 4 23.512 0.303 23.039 0.141 23.037 0.233 23.055 0.488 22.985 0.598
## 5 24.159 0.402 23.308 0.130 23.490 0.266 22.744 0.271 23.488 0.708
## 6 25.667 0.563 26.351 0.624 24.877 0.391 25.389 1.205 24.428 0.709
## J dJ H dH KS dKS F814W dF814W F814W_3arcs
## 1 23.283 0.160 23.130 0.231 23.223 0.593 23.977 0.136 23.999
## 2 24.380 0.354 24.382 0.596 99.000 22.750 25.339 0.780 25.476
## 3 24.374 0.460 99.000 24.170 99.000 22.750 24.420 0.295 24.535
## 4 22.442 0.110 22.385 0.166 21.847 0.246 23.230 0.107 23.447
## 5 22.682 0.105 22.559 0.148 22.099 0.237 23.430 0.094 23.464
## 6 26.811 1.716 99.000 24.170 99.000 22.750 25.641 0.249 26.072
## dF814W_3arcs F814W_3arcs_corr nfobs xray PercW Satur_Flag Stellar_Flag
## 1 0.276 24.101 22 0 0.831 0 0.5
## 2 0.710 25.004 14 0 0.829 0 0.5
## 3 0.385 24.592 16 0 0.946 0 0.5
## 4 0.193 23.568 24 0 0.859 0 0.5
## 5 0.193 23.583 24 0 0.854 0 0.5
## 6 1.091 25.068 16 0 0.836 0 0.5
## zb_1 zb_min_1 zb_max_1 tb_1 Odds_1 Chi2 Stell_Mass_1 M_ABS_1
## 1 0.807 0.749 0.915 7.074 0.229 0.875 9.461 -18.749
## 2 1.438 0.373 2.546 7.351 0.022 0.839 9.685 -19.418
## 3 0.988 0.219 1.054 7.377 0.186 1.158 9.439 -18.858
## 4 1.040 1.030 1.116 7.381 0.449 0.957 9.987 -20.229
## 5 1.343 1.310 1.467 8.838 0.460 0.977 9.880 -20.759
## 6 1.036 0.700 1.835 7.382 0.145 0.650 9.017 -17.804
## irms_OPT_Flag irms_NIR_Flag
## 1 0 0
## 2 0 0
## 3 0 0
## 4 0 0
## 5 0 0
## 6 0 0

```

Descripción de los datos

```
describe(data)
```

```

## data
##
## 86 Variables     446343 Observations
## -----
## RA
##      n missing distinct      Info      Mean      Gmd      .05      .10
## 446343      0    80684       1      194     109.8     36.98     37.29
##      .25      .50      .75      .90      .95
## 139.11   189.74   243.18   356.37   356.66
##
## Value      35     40     140    150     190     215     240     245     355
## Frequency 63764 11830 68045 38427 42494 66974 9937 69730 75142
## Proportion 0.143 0.027 0.152 0.086 0.095 0.150 0.022 0.156 0.168
## -----

```

```

## DEC
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343       0    34644       1   33.54   25.58   0.7031   1.0850
##  .25       .50     .75     .90     .95
##  2.5396  46.2961  54.0902  54.7335  61.9446
##
##      Value      0.5    1.0    1.5    2.0    2.5   15.5   16.0   46.0   46.5   52.0
##      Frequency  29006  44309  2279  17457  20970  35254  39888  30577  37468  10585
##      Proportion 0.065  0.099  0.005  0.039  0.047  0.079  0.089  0.069  0.084  0.024
##
##      Value      52.5   53.0   54.0   54.5   61.5   62.0   62.5
##      Frequency  35121  21268  36263  43404  3673   22781  16040
##      Proportion 0.079  0.048  0.081  0.097  0.008  0.051  0.036
## -----
## dis
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343       0    342401       1   318522   175597   57181   58530
##  .25       .50     .75     .90     .95
##  133833  380333  434151  439044  539690
##
##      Value      55000  60000  130000  135000  375000  380000  420000  425000  430000
##      Frequency  31041  44101   8320   67274   5340   74327   14460   28034   25728
##      Proportion 0.070  0.099  0.019  0.151  0.012  0.167  0.032  0.063  0.058
##
##      Value      435000  440000  540000
##      Frequency  73466   35825   38427
##      Proportion 0.165  0.080  0.086
## -----
## objID
##      n  missing distinct      Info      Mean      Gmd      .05
##  446343       0    438661       1  8.145e+10  24809865  8.142e+10
##  .10       .25     .50     .75     .90     .95
##  8.142e+10  8.143e+10  8.145e+10  8.147e+10  8.148e+10  8.148e+10
##
##      lowest : 81421100001 81421100183 81421100216 81421100222 81421100234
##      highest: 81482410412 81482410415 81482410416 81482410417 81482410419
## -----
## Field
##      n  missing distinct      Info      Mean      Gmd
##  446343       0        7   0.976   5.113   2.434
##
##      Value      2      3      4      5      6      7      8
##      Frequency  75594  68045  38427  42494  66974  79667  75142
##      Proportion 0.169  0.152  0.086  0.095  0.150  0.178  0.168
## -----
## Pointing
##      n  missing distinct      Info      Mean      Gmd
##  446343       0        4   0.831   1.73   0.9246
##
##      Value      1      2      3      4
##      Frequency  231970 134706  47928  31739
##      Proportion 0.520  0.302  0.107  0.071
## -----
## CCD

```

```

##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343       0        4     0.936     2.444     1.267
##
## Value         1      2      3      4
## Frequency   120606 118225  96023 111489
## Proportion  0.270  0.265  0.215  0.250
## -----
## x
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343       0    415339       1     2510     1312    743.3    937.0
##  .25       .50       .75       .90       .95
##  1525.2   2508.3   3497.7   4081.8   4280.3
##
## lowest : 488.566 488.694 491.682 491.864 492.180
## highest: 4531.101 4532.417 4534.300 4534.376 4535.785
## -----
## y
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343       0    415100       1     2514     1318    731.9    930.4
##  .25       .50       .75       .90       .95
##  1525.6   2522.8   3504.4   4089.3   4285.5
##
## lowest : 451.343 454.649 457.413 458.174 458.604
## highest: 4544.644 4545.848 4546.638 4548.665 4549.844
## -----
## area
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343       0      3627       1    112.8    156.8      8      11
##  .25       .50       .75       .90       .95
##  18       39       93      214      370
##
## lowest : 2      3      4      5      6, highest: 29090 29529 36069 43872 58150
## -----
## fwhm
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343       0      3853       1     8.459     3.254     4.65     5.22
##  .25       .50       .75       .90       .95
##  6.42     7.92     9.76    11.82    13.41
##
## lowest : -3.32 -1.22 -0.59 -0.47 -0.36, highest: 219.04 252.26 269.52 284.63 292.08
## -----
## stell
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343       0      101      0.999     0.4143     0.318     0.02     0.03
##  .25       .50       .75       .90       .95
##  0.14     0.46     0.60      0.78      0.94
##
## lowest : 0.00 0.01 0.02 0.03 0.04, highest: 0.96 0.97 0.98 0.99 1.00
## -----
## ell
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343       0      7872       1     0.26     0.1704    0.0532    0.0787
##  .25       .50       .75       .90       .95
##  0.1402   0.2379   0.3592    0.4757    0.5426

```

```

##
## lowest : 0.0004 0.0008 0.0009 0.0012 0.0013, highest: 0.9490 0.9506 0.9636 0.9652 0.9721
## -----
## a
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     9492       1    2.449    1.182    1.190    1.350
##  .25      .50      .75      .90      .95
##  1.683    2.171    2.844    3.709    4.427
##
## lowest : 0.501 0.502 0.503 0.504 0.505
## highest: 80.280 86.029 107.074 140.702 240.707
## -----
## b
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     7185       1    1.81    0.9806   0.718    0.837
##  .25      .50      .75      .90      .95
##  1.131    1.631    2.263    2.937    3.434
##
## lowest : 0.211 0.284 0.288 0.289 0.301, highest: 25.665 25.843 28.092 34.853 35.888
## -----
## theta
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     1801       1    2.087    55.06   -78.3   -66.7
##  .25      .50      .75      .90      .95
##  -35.3     3.7     39.8     67.9     78.7
##
## lowest : -90.0 -89.9 -89.8 -89.7 -89.6, highest: 89.6 89.7 89.8 89.9 90.0
## -----
## rk
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     647     0.973    4.794    1.482    3.50     3.50
##  .25      .50      .75      .90      .95
##  3.50     4.39     5.77     6.86     7.41
##
## lowest : 3.50 3.51 3.52 3.53 3.54, highest: 12.76 12.77 13.03 13.47 14.19
## -----
## rf
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    10628       1    3.546    1.374    2.042    2.282
##  .25      .50      .75      .90      .95
##  2.710    3.238    3.951    4.859    5.656
##
## Value      -400     -100       0     100     200     300     600   10900
## Frequency     1        2 446271      59       6       2       1       1
## Proportion    0        0     1       0       0       0       0       0
## -----
## s2n
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    32618       1    91.5    161.9     4.05     4.62
##  .25      .50      .75      .90      .95
##  6.12     9.84    21.76    65.59   152.26
##
## lowest : 3.00 3.01 3.02 3.03 3.04
## highest: 28821.18 29183.57 30230.37 31198.24 50028.99

```

```

## -----
## photoflag
##      n  missing distinct      Info      Mean      Gmd
##  446343      0       9    0.711    0.8443    1.176
##
## Value      0     1     2     3     4     5     6     7     19
## Frequency 291980 6785 73640 73582   86     1    232    35     2
## Proportion 0.654 0.015 0.165 0.165 0.000 0.000 0.001 0.000 0.000
## -----
## F365W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    9876    0.971    47.26    32.85    22.84    23.52
##  .25      .50     .75     .90     .95
##  24.39    25.29    99.00    99.00    99.00
##
## Value     -100     12     14     16     18     20     22     24     26
## Frequency 925      1     56     525    1132    3288  20141  160994 119732
## Proportion 0.002 0.000 0.000 0.001 0.003 0.007 0.045 0.361 0.268
##
## Value      28     30     100
## Frequency 1984      3 137562
## Proportion 0.004 0.000 0.308
## -----
## dF365W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    3236      1    8.212    10.83    0.113    0.187
##  .25      .50     .75     .90     .95
##  0.368    0.756   24.780   25.600   25.800
##
## lowest :  0.000 0.001 0.002 0.003 0.004, highest: 26.000 27.060 27.140 27.660 27.750
## -----
## F396W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0   10414    0.991    39.9     25.2    22.71    23.43
##  .25      .50     .75     .90     .95
##  24.33    25.22   26.38    99.00    99.00
##
## Value     -100     12     14     16     18     20     22     24     26
## Frequency 177      1     95     656    1293    4060  22761  164935 156093
## Proportion 0.000 0.000 0.000 0.001 0.003 0.009 0.051 0.370 0.350
##
## Value      28     100
## Frequency 4701  91571
## Proportion 0.011 0.205
## -----
## dF396W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    2639      1    5.631     8.54    0.060    0.095
##  .25      .50     .75     .90     .95
##  0.200    0.436   1.103   25.800   25.920
##
## lowest :  0.000 0.001 0.002 0.003 0.004, highest: 27.470 27.500 27.590 27.630 28.670
## -----
## F427W

```

```

##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    10749    0.995    36.97    21.55    22.51    23.29
##  .25       .50     .75     .90     .95
##  24.24    25.13    26.16    99.00    99.00
##
##          Value     -100      12      14      16      18      20      22      24      26
##          Frequency   22       3     162     818    1629    5054   26256  171781 160747
##          Proportion 0.000    0.000    0.000    0.002    0.004    0.011   0.059   0.385   0.360
##
##          Value      28      30     100
##          Frequency  6038      1   73832
##          Proportion 0.014    0.000    0.165
## -----
##          dF427W
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    2621      1    4.606    7.297    0.059    0.089
##  .25       .50     .75     .90     .95
##  0.171    0.367    0.877    25.790   26.070
##
##          lowest :  0.000  0.001  0.002  0.003  0.004, highest: 27.380 27.400 27.420 27.500 32.650
## -----
##          F458W
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    11079    0.997    35.06   19.93    22.20    23.11
##  .25       .50     .75     .90     .95
##  24.13    25.04    25.98    99.00    99.00
##
##          Value     -100      12      14      16      18      20      22      24      26
##          Frequency  877       3     241    1009    2010    6365   30408  177658 159623
##          Proportion 0.002    0.000    0.001    0.002    0.005    0.014   0.068   0.398   0.358
##
##          Value      28      100
##          Frequency  3862   64287
##          Proportion 0.009    0.144
## -----
##          dF458W
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    2745      1    4.089    6.554    0.062    0.100
##  .25       .50     .75     .90     .95
##  0.197    0.388    0.828    25.700   26.010
##
##          lowest :  0.000  0.001  0.002  0.003  0.004, highest: 27.390 27.440 27.490 27.560 27.710
## -----
##          F489W
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    11295    0.999    31.43   13.82    22.05    22.98
##  .25       .50     .75     .90     .95
##  24.04    24.95    25.83    27.17    99.00
##
##          Value     -100      12      14      16      18      20      22      24      26
##          Frequency  101       3     247    1083    2192    7488   34174  185396 168765
##          Proportion 0.000    0.000    0.001    0.002    0.005    0.017   0.077   0.415   0.378
##
##          Value      28      100

```

```

## Frequency      5655  41239
## Proportion   0.013  0.092
##
## dF489W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343        0     2229        1    2.723    4.603    0.039    0.065
##  .25       .50       .75       .90       .95
##  0.131     0.262     0.541    1.353   26.220
##
## lowest :  0.000  0.001  0.002  0.003  0.004, highest: 27.560 27.630 27.690 27.790 27.840
##
## F520W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343        0     11517      0.999    30.59    13.22    21.79    22.80
##  .25       .50       .75       .90       .95
##  23.94     24.86     25.73    26.86   99.00
##
## Value      -100      12      14      16      18      20      22      24      26
## Frequency    684       3     289     1127     2479    8991   38194  191256 160929
## Proportion  0.002  0.000  0.001  0.003  0.006  0.020  0.086  0.428  0.361
##
## Value      28       30      100
## Frequency   4737       7   37647
## Proportion  0.011  0.000  0.084
##
## dF520W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343        0     2266        1    2.496    4.243    0.034    0.058
##  .25       .50       .75       .90       .95
##  0.122     0.250     0.517    1.217   26.040
##
## lowest :  0.000  0.001  0.002  0.003  0.004, highest: 27.330 27.360 27.530 27.730 28.090
##
## F551W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343        0     11509      0.999    32.78    16.3    21.64    22.68
##  .25       .50       .75       .90       .95
##  23.84     24.76     25.66    99.00   99.00
##
## Value      -100      12      14      16      18      20      22      24      26
## Frequency    56       4     306     1294     2783   10068  42204  199191 136347
## Proportion  0.000  0.000  0.001  0.003  0.006  0.023  0.095  0.446  0.305
##
## Value      28       30      100
## Frequency   3491       2   50597
## Proportion  0.008  0.000  0.113
##
## dF551W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343        0     2339        1    3.21    5.352    0.034    0.059
##  .25       .50       .75       .90       .95
##  0.128     0.287     0.623   24.980   25.690
##
## lowest :  0.000  0.001  0.002  0.003  0.004, highest: 27.240 27.390 27.400 27.480 27.690

```

```

## -----
## F582W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    11440     0.999    31.31    15.76    21.38    22.49
##  .25      .50      .75      .90      .95
##  23.73    24.67    25.54    99.00    99.00
##
## Value      -100      12      14      16      18      20      22      24      26
## Frequency   1686      3     333    1357    2959   11318   46047  205573 130214
## Proportion  0.004    0.000   0.001   0.003   0.007   0.025   0.103   0.461   0.292
##
## Value      28      100
## Frequency  1910  44943
## Proportion 0.004    0.101
##
## -----
## dF582W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    2056       1    2.885    4.848    0.033    0.059
##  .25      .50      .75      .90      .95
##  0.131    0.276    0.571   24.620   25.780
##
## lowest :  0.000  0.001  0.002  0.003  0.004, highest: 27.020 27.130 27.260 27.330 27.500
##
## -----
## F613W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    11877       1    28.36    9.403    21.34    22.39
##  .25      .50      .75      .90      .95
##  23.65    24.61    25.45   26.32    99.00
##
## Value      -100      12      14      16      18      20      22      24      26
## Frequency   159       6     319    1453    3180   12694   50713  209568 139562
## Proportion  0.000    0.000   0.001   0.003   0.007   0.028   0.114   0.470   0.313
##
## Value      28      100
## Frequency  3966  24723
## Proportion 0.009    0.055
##
## -----
## dF613W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    1911       1    1.667    2.925    0.019    0.034
##  .25      .50      .75      .90      .95
##  0.072    0.159    0.354   0.776   25.580
##
## lowest :  0.000  0.001  0.002  0.003  0.004, highest: 27.540 27.610 27.650 27.710 27.910
##
## -----
## F644W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    11985       1    27.16    7.464    21.15    22.22
##  .25      .50      .75      .90      .95
##  23.54    24.54    25.37   26.18   26.98
##
## Value      -100      12      14      16      18      20      22      24      26
## Frequency   159       5     336    1578    3628   14545   55768  211336 136864
## Proportion  0.000    0.000   0.001   0.004   0.008   0.033   0.125   0.473   0.307

```

```

## 
## Value      28    100
## Frequency 4145 17979
## Proportion 0.009 0.040
## -----
## dF644W
##      n  missing distinct     Info      Mean      Gmd      .05      .10
## 446343      0    1985       1    1.294    2.223    0.024    0.042
##   .25      .50      .75      .90      .95
##   0.091    0.186    0.356    0.675    1.171
## 
## lowest :  0.000  0.001  0.002  0.003  0.004, highest: 27.580 27.630 27.720 27.770 27.780
## -----
## F675W
##      n  missing distinct     Info      Mean      Gmd      .05      .10
## 446343      0    11927       1    26.52    6.588    21.05    22.11
##   .25      .50      .75      .90      .95
##   23.44    24.46    25.30    26.07    26.70
## 
## Value      -100     12     14     16     18     20     22     24     26
## Frequency   317      4    395    1557    3721   15559   60253  215335 131190
## Proportion  0.001  0.000  0.001  0.003  0.008  0.035  0.135  0.482  0.294
## 
## Value      28    100
## Frequency 3249 14763
## Proportion 0.007 0.033
## -----
## dF675W
##      n  missing distinct     Info      Mean      Gmd      .05      .10
## 446343      0    1972       1    1.085    1.858    0.024    0.040
##   .25      .50      .75      .90      .95
##   0.082    0.167    0.327    0.610    1.010
## 
## lowest :  0.000  0.001  0.002  0.003  0.004, highest: 27.490 27.690 27.720 27.770 27.850
## -----
## F706W
##      n  missing distinct     Info      Mean      Gmd      .05      .10
## 446343      0    11984       1    25.29    4.286    20.94    21.98
##   .25      .50      .75      .90      .95
##   23.34    24.37    25.18    25.86    26.32
## 
## Value      -100     12     14     16     18     20     22     24     26
## Frequency  143      9    426    1648    4101   16944   65270  223233 124420
## Proportion 0.000  0.000  0.001  0.004  0.009  0.038  0.146  0.500  0.279
## 
## Value      28     30     100
## Frequency 2504     1    7644
## Proportion 0.006  0.000  0.017
## -----
## dF706W
##      n  missing distinct     Info      Mean      Gmd      .05      .10
## 446343      0    1886       1    0.6491   1.064    0.021    0.034
##   .25      .50      .75      .90      .95
##   0.070    0.145    0.277    0.477    0.697

```

```

## 
## lowest : 0.000 0.001 0.002 0.003 0.004, highest: 27.690 27.740 27.770 27.800 27.840
## -----
## F737W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    11900       1    25.14    4.218    20.79    21.84
##  .25      .50      .75      .90      .95
##  23.22    24.26    25.05    25.70    26.13
## 
## Value     -100      12      14      16      18      20      22      24      26
## Frequency   128       3     455    1786    4722   18444   70637  232405 109046
## Proportion 0.000  0.000  0.001  0.004  0.011  0.041  0.158  0.521  0.244
## 
## Value      28      100
## Frequency  1250    7467
## Proportion 0.003  0.017
## -----
## dF737W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    1810       1    0.6543   1.042    0.023  0.040
##  .25      .50      .75      .90      .95
##  0.085    0.170    0.313    0.515    0.729
## 
## lowest : 0.000 0.001 0.002 0.003 0.004, highest: 27.300 27.360 27.440 27.590 27.650
## -----
## F768W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    11664       1    24.83    4.939    20.64    21.71
##  .25      .50      .75      .90      .95
##  23.11    24.16    24.93    25.56    25.97
## 
## Value     -100      12      14      16      18      20      22      24      26
## Frequency  1170       1     491    1779    5012   19510   76139  239583 94152
## Proportion 0.003  0.000  0.001  0.004  0.011  0.044  0.171  0.537  0.211
## 
## Value      28      100
## Frequency  493    8013
## Proportion 0.001  0.018
## -----
## dF768W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    1816       1    0.6929   1.107    0.021  0.038
##  .25      .50      .75      .90      .95
##  0.085    0.181    0.331    0.542    0.768
## 
## lowest : 0.000 0.001 0.002 0.003 0.004, highest: 27.330 27.340 27.450 27.510 27.580
## -----
## F799W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    11837       1    24.94    4.084    20.62    21.65
##  .25      .50      .75      .90      .95
##  23.02    24.09    24.89    25.54    25.96
## 
## Value     -100      12      14      16      18      20      22      24      26

```

```

## Frequency      22      9     503    1885    5338   20912   81454  238953  89394
## Proportion   0.000  0.000  0.001  0.004  0.012  0.047  0.182  0.535  0.200
##
## Value         28     100
## Frequency    709    7164
## Proportion  0.002  0.016
##
## -----
## dF799W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     1705        1  0.6166  0.9975  0.018  0.032
##  .25       .50     .75        .90     .95
##  0.070     0.148   0.293      0.492     0.695
##
## lowest :  0.000  0.001  0.002  0.003  0.004, highest: 27.300 27.350 27.400 27.420 27.650
## -----
## F830W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     11622        1  26.17   6.579  20.54  21.57
##  .25       .50     .75        .90     .95
##  22.94     24.02   24.83      25.52     26.07
##
## Value        -100     12      14      16      18      20      22      24      26
## Frequency     39      9     551    1940    5668   22202   85502  239525  75510
## Proportion  0.000  0.000  0.001  0.004  0.013  0.050  0.192  0.537  0.169
##
## Value         28     100
## Frequency    281   15116
## Proportion  0.001  0.034
##
## -----
## dF830W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     1820        1  1.133   1.873  0.028  0.050
##  .25       .50     .75        .90     .95
##  0.114     0.233   0.407      0.688     1.039
##
## lowest :  0.000  0.001  0.002  0.003  0.004, highest: 26.490 26.540 26.590 26.790 26.830
## -----
## F861W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     11613        1  25.43   5.307  20.48  21.50
##  .25       .50     .75        .90     .95
##  22.85     23.93   24.75      25.42     25.89
##
## Value        -100     12      14      16      18      20      22      24      26
## Frequency     18      9     577    1942    5848   23315   91118  242027  70210
## Proportion  0.000  0.000  0.001  0.004  0.013  0.052  0.204  0.542  0.157
##
## Value         28     100
## Frequency    248   11031
## Proportion  0.001  0.025
##
## -----
## dF861W
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     1801        1  0.8671  1.419  0.026  0.045

```

```

##      .25      .50      .75      .90      .95
##  0.096   0.192   0.351   0.592   0.873
##
## lowest :  0.000  0.001  0.002  0.003  0.004, highest: 26.680 26.700 26.890 26.900 27.030
## -----
## F892W
##      n missing distinct      Info      Mean      Gmd      .05      .10
##  446343     0    11275         1    28.68    11.37    20.42    21.43
##      .25      .50      .75      .90      .95
##  22.75   23.78   24.58   25.41   99.00
##
## Value      -100      12      14      16      18      20      22      24      26
## Frequency     9      13     601    2029     6089   24466   99172  245355  37054
## Proportion  0.000  0.000  0.001  0.005  0.014  0.055  0.222  0.550  0.083
##
## Value      28      100
## Frequency    38    31517
## Proportion  0.000  0.071
## -----
## dF892W
##      n missing distinct      Info      Mean      Gmd      .05      .10
##  446343     0    1801         1    2.062    3.442    0.039  0.069
##      .25      .50      .75      .90      .95
##  0.150   0.305   0.548   1.021   24.360
##
## lowest :  0.000  0.001  0.002  0.003  0.004, highest: 25.930 26.020 26.110 26.170 26.560
## -----
## F923W
##      n missing distinct      Info      Mean      Gmd      .05      .10
##  446343     0    11230        0.998    31.9    16.76    20.36    21.38
##      .25      .50      .75      .90      .95
##  22.70   23.71   24.56   99.00   99.00
##
## Value      -100      12      14      16      18      20      22      24      26
## Frequency    52      25     605    2105     6464   25261  103991  232725  23720
## Proportion  0.000  0.000  0.001  0.005  0.014  0.057  0.233  0.521  0.053
##
## Value      28      100
## Frequency    22    51373
## Proportion  0.000  0.115
## -----
## dF923W
##      n missing distinct      Info      Mean      Gmd      .05      .10
##  446343     0    2228         1    3.142    5.073    0.059  0.099
##      .25      .50      .75      .90      .95
##  0.207   0.414   0.764   23.460   24.020
##
## lowest :  0.000  0.001  0.002  0.003  0.004, highest: 25.480 25.640 25.700 25.930 26.100
## -----
## F954W
##      n missing distinct      Info      Mean      Gmd      .05      .10
##  446343     0    10602        0.997    33.97   20.05    20.38    21.34
##      .25      .50      .75      .90      .95
##  22.48   23.34   24.17   99.00   99.00

```

```

## 
## Value      12     13     14     15     16     17     18     19     20
## Frequency  11     73    277    590    984   1616   2906   5856  11856
## Proportion 0.000  0.000  0.001  0.001  0.002  0.004  0.007  0.013 0.027
## 
## Value      21     22     23     24     25     26     27     99
## Frequency 26189 62576 133984 111939 20982 1366    1 65137
## Proportion 0.059 0.140 0.300 0.251 0.047 0.003 0.000 0.146
## -----
## dF954W
##      n missing distinct      Info      Mean      Gmd      .05      .10
## 446343      0     2193          1      3.78     5.938     0.080     0.133
##  .25       .50       .75       .90       .95
## 0.252     0.462     0.864    22.910    23.390
## 
## lowest : 0.001 0.002 0.003 0.004 0.005, highest: 25.040 25.050 25.060 25.200 27.177
## -----
## J
##      n missing distinct      Info      Mean      Gmd      .05      .10
## 446343      0     11905          1     28.36    13.21    19.89    20.94
##  .25       .50       .75       .90       .95
## 22.33     23.51     24.53    25.74    99.00
## 
## Value      -100    12     14     16     18     20     22     24     26
## Frequency 1805     50    659    2355   8019   33647  122062  202302  41387
## Proportion 0.004  0.000  0.001  0.005  0.018  0.075  0.273  0.453  0.093
## 
## Value      28     100
## Frequency 198   33859
## Proportion 0.000  0.076
## -----
## dJ
##      n missing distinct      Info      Mean      Gmd      .05      .10
## 446343      0     2486          1     2.187    3.726    0.0200   0.0390
##  .25       .50       .75       .90       .95
## 0.1010    0.2420    0.5315   1.4380   24.4800
## 
## lowest : 0.000 0.001 0.002 0.003 0.004, highest: 25.680 25.690 25.700 25.930 26.200
## -----
## H
##      n missing distinct      Info      Mean      Gmd      .05      .10
## 446343      0     11539        0.998    31.6    18.2    19.66    20.64
##  .25       .50       .75       .90       .95
## 22.03     23.25     24.37    99.00    99.00
## 
## Value      -100    12     14     16     18     20     22     24     26
## Frequency 1141     44    656    2589   9785   42914  138927  175506  21043
## Proportion 0.003  0.000  0.001  0.006  0.022  0.096  0.311  0.393  0.047
## 
## Value      28     100
## Frequency 15    53723
## Proportion 0.000  0.120
## -----
## dH

```

```

##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343       0     2554        1    3.246    5.337    0.026    0.048
##  .25       .50     .75     .90     .95
##  0.127     0.316    0.755   23.690   24.030
##
## lowest :  0.000  0.001  0.002  0.003  0.004, highest: 25.020 25.080 25.130 25.200 25.240
## -----
## KS
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343       0     10905       0.994    36.29    24.65    19.57    20.48
##  .25       .50     .75     .90     .95
##  21.89     23.14    24.47    99.00    99.00
##
## Value      -100      12      14      16      18      20      22      24      26
## Frequency    892      22     435    2380   10652   49592  144471  145691  10203
## Proportion  0.002  0.000  0.001  0.005  0.024  0.111  0.324  0.326  0.023
##
## Value       28      100
## Frequency     1  82004
## Proportion  0.000  0.184
## -----
## dKS
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343       0     2537        1    4.697    7.265    0.030    0.055
##  .25       .50     .75     .90     .95
##  0.155     0.407    1.196   23.400   24.100
##
## lowest :  0.000  0.001  0.002  0.003  0.004, highest: 24.440 24.500 24.550 24.610 24.680
## -----
## F814W
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343       0     10869       1    23.62    1.646    20.58    21.60
##  .25       .50     .75     .90     .95
##  22.94     23.96    24.69    25.20    25.46
##
## lowest : 12.981 13.142 13.165 13.167 13.170, highest: 26.735 26.762 26.772 26.780 26.942
## -----
## dF814W
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343       0     1700        1   0.1674    0.126    0.023    0.040
##  .25       .50     .75     .90     .95
##  0.083     0.147    0.218    0.296    0.367
##
## lowest : 0.000 0.001 0.002 0.003 0.004, highest: 5.833 5.899 6.232 6.495 6.546
## -----
## F814W_3arcs
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343       0     11010       1    24.06     2.1    20.89    21.86
##  .25       .50     .75     .90     .95
##  23.16     24.16    24.79    25.25    25.53
##
## Value       13      14      15      16      17      18      19      20      21
## Frequency     4     175     522     807    1224    2082    3890    8250  17538
## Proportion  0.000  0.000  0.001  0.002  0.003  0.005  0.009  0.018  0.039

```

```

## 
## Value      22     23     24     25     26     27     99
## Frequency 35722  70710 139730 141735 21392   928   1634
## Proportion 0.080  0.158 0.313  0.318  0.048  0.002  0.004
## -----
## df814W_3arcs
##      n missing distinct      Info      Mean      Gmd      .05      .10
## 446343      0     1680        1    0.3253    0.3856    0.016    0.035
##  .25       .50       .75       .90       .95
##  0.091     0.193    0.318     0.460     0.575
## 
## Value      0.0     0.2     0.4     0.6     0.8     1.0     1.2     1.4     1.6
## Frequency 122277 199693 89818  22654  6277  2247  967  532  207
## Proportion 0.274  0.447  0.201  0.051  0.014  0.005  0.002  0.001  0.000
## 
## Value      1.8     2.0     2.2    26.4    26.6    26.8    27.0    27.2
## Frequency  32      4       1       8      562    752    237     75
## Proportion 0.000  0.000  0.000  0.000  0.001  0.002  0.001  0.000
## -----
## F814W_3arcs_corr
##      n missing distinct      Info      Mean      Gmd      .05      .10
## 446343      0     9719        1    24.07    1.893   21.08   22.04
##  .25       .50       .75       .90       .95
##  23.30    24.26    24.78    24.97    25.01
## 
## Value      11      12      13      14      15      16      17      18      19
## Frequency  2       20     198     183     183     740    1138    1872   3451
## Proportion 0.000  0.000  0.000  0.000  0.000  0.002  0.003  0.004  0.008
## 
## Value      20      21      22      23      24      25      26      97      98
## Frequency 7088  15430  32027  67144  137985  176555  693    173    1461
## Proportion 0.016  0.035  0.072  0.150  0.309  0.396  0.002  0.000  0.003
## -----
## nfobs
##      n missing distinct      Info      Mean      Gmd      .05      .10
## 446343      0     24       0.933    21.78    2.842    16      18
##  .25       .50       .75       .90       .95
##  21       23       24       24       24
## 
## lowest :  1  2  3  4  5, highest: 20 21 22 23 24
## -----
## xray
##      n missing distinct      Info      Sum      Mean      Gmd
## 446343      0       2        0      55  0.0001232 0.0002464
## 
## -----
## PercW
##      n missing distinct      Info      Mean      Gmd      .05      .10
## 446343      0     201      0.99  0.9792  0.02476    0.927    0.952
##  .25       .50       .75       .90       .95
##  0.974    0.987    0.996     1.000    1.000
## 
## lowest : 0.800 0.801 0.802 0.803 0.804, highest: 0.996 0.997 0.998 0.999 1.000
## -----

```

```

## Satur_Flag
##      n  missing distinct      Info      Sum      Mean      Gmd
##  446343      0       2    0.008    1185 0.002655 0.005296
##
## -----
## Stellar_Flag
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     101    0.459    0.4562   0.1496      0.00    0.03
##  .25       .50       .75      .90      .95
##  0.50       0.50       0.50      0.50      0.50
##
## lowest : 0.00 0.01 0.02 0.03 0.04, highest: 0.96 0.97 0.98 0.99 1.00
## -----
## zb_1
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     4251      1    0.8482   0.5875    0.164    0.240
##  .25       .50       .75      .90      .95
##  0.433      0.785     1.128    1.514    1.792
##
## lowest : 0.003 0.007 0.009 0.010 0.011, highest: 6.640 6.658 6.672 6.693 6.726
## -----
## zb_min_1
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     3929      1    0.6603   0.5208    0.086    0.130
##  .25       .50       .75      .90      .95
##  0.268      0.582     0.947    1.281    1.462
##
## lowest : 0.003 0.005 0.006 0.007 0.008, highest: 6.573 6.590 6.591 6.601 6.634
## -----
## zb_max_1
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     5033      1    1.065    0.7142    0.244    0.333
##  .25       .50       .75      .90      .95
##  0.573      0.945     1.381    1.964    2.286
##
## lowest : 0.016 0.023 0.024 0.025 0.026, highest: 6.806 6.847 6.906 6.953 6.956
## -----
## tb_1
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     9991      1     7.23    1.964    3.611   4.592
##  .25       .50       .75      .90      .95
##  6.582     7.338     8.303    9.355   10.241
##
## lowest : 1.002 1.003 1.004 1.005 1.006, highest: 10.993 10.994 10.995 10.996 10.998
## -----
## Odds_1
##      n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     1001      1    0.3348   0.2996    0.036    0.052
##  .25       .50       .75      .90      .95
##  0.100     0.249     0.538    0.752    0.854
##
## lowest : 0.000 0.001 0.002 0.003 0.004, highest: 0.996 0.997 0.998 0.999 1.000
## -----
## Chi2

```

```

##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     9058       1   0.9295   0.8145   0.268   0.325
##  .25      .50      .75      .90      .95
##  0.445    0.630    0.905    1.335    1.988
##
## lowest :  0.002  0.007  0.010  0.013  0.015, highest: 98.372 98.526 98.752 98.850 99.000
## -----
## Stell_Mass_1
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0     5982       1   9.341    1.062    7.733   8.064
##  .25      .50      .75      .90      .95
##  8.712    9.394   10.018   10.521   10.790
##
## lowest :  4.416  4.566  4.658  4.693  4.743, highest: 12.646 12.677 12.687 12.688 12.781
## -----
## M_ABS_1
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0    12844       1  -18.63    2.414   -21.64  -21.16
##  .25      .50      .75      .90      .95
##  -20.25   -18.90   -17.19   -15.64   -14.82
##
## lowest : -28.285 -27.620 -27.557 -27.391 -27.156
## highest: -9.333 -9.255 -9.250 -9.115 -8.513
## -----
## irms_OPT_Flag
##          n  missing distinct      Info      Mean      Gmd      .05      .10
##  446343      0       21       0.13   0.1033   0.2014      0       0
##  .25      .50      .75      .90      .95
##  0       0       0       0       0
##
## lowest :  0  1  2  3  4, highest: 16 17 18 19 21
## -----
## irms_NIR_Flag
##          n  missing distinct      Info      Mean      Gmd
##  446343      0       4       0.298   0.142   0.2579
##
## Value      0       1       2       3
## Frequency 396517 39505 7087 3234
## Proportion 0.888 0.089 0.016 0.007
## -----

```

### 3 Limpieza inicial y selección de datos útiles

Selección de datos útiles para el estudio, junto con variables útiles en la medida de calidad para revisión a posteriori si es necesario.

*#Selección de variables: generamos dos ficheros, uno que conserva las variables útiles para el objetivo del estudio y ademas variables de calidad que pueden proporcionar informacion a posteriori si es necesario y otro con solo las variables utiles:*

```
util_data_1 <- data %>% select(RA, DEC, objID, stell,s2n, F365W:dF814W,
                                    nfobs, Satur_Flag, Stellar_Flag, zb_1, M_ABS_1)
```

```

#filtrado inicial: eliminacion de los datos con saturacion y de los datos con
#z muy grande
datosmod = util_data_1 %>% filter(zb_1<0.5 & Satur_Flag==0)

#ordenacion de variables
data_util <- datosmod %>% select(objID,RA, DEC, stell, matches("^F.*W$"),
                                    J, H, KS, starts_with("d"),Stellar_Flag, M_ABS_1)

#seleccion final de las variables para usar en el analisis
util_data <- data_util %>% select(objID:F954W,J,H,KS,F814W,Stellar_Flag)
head(util_data)

##          objID        RA      DEC stell F365W  F396W  F427W  F458W  F489W
## 1 81481409774 356.9394 15.3444  0.64 99.000 25.144 25.009 24.800 23.423
## 2 81481409790 356.9355 15.3440  0.40 99.000 99.000 99.000 99.000 99.000
## 3 81481409773 356.9335 15.3442  0.45 99.000 24.630 99.000 99.000 26.061
## 4 81481409588 356.9476 15.3481  0.99 19.165 18.910 18.657 18.377 18.252
## 5 81481409853 356.9159 15.3424  0.91 99.000 99.000 99.000 99.000 24.965
## 6 81481409764 356.9278 15.3448  0.36 99.000 99.000 99.000 25.251 99.000
##          F520W  F551W  F582W  F613W  F644W  F675W  F706W  F737W  F768W  F799W
## 1 24.081 24.125 24.477 24.255 24.204 23.824 23.884 23.990 24.260 23.526
## 2 99.000 25.804 25.983 25.226 25.196 24.906 25.520 99.000 25.313 25.615
## 3 26.050 24.805 26.052 24.868 26.167 26.212 25.639 99.000 24.586 24.888
## 4 18.138 18.211 18.117 18.106 17.994 17.985 17.862 17.752 17.786 17.804
## 5 24.862 24.442 23.908 23.570 23.510 23.547 23.170 22.922 22.826 22.580
## 6 24.530 24.771 24.832 24.628 24.559 24.406 24.442 24.486 23.884 23.943
##          F830W  F861W  F892W  F923W  F954W        J        H        KS  F814W
## 1 23.554 23.972 23.575 23.178 22.678 23.971 24.670 24.007 23.498
## 2 99.000 25.202 24.544 99.000 23.986 24.609 99.000 99.000 25.252
## 3 24.731 25.164 24.747 24.295 99.000 25.071 24.593 99.000 24.973
## 4 17.671 17.763 17.713 17.626 17.804 17.770 17.959 18.328 17.609
## 5 22.675 22.439 22.230 22.081 22.230 21.992 21.782 22.274 22.473
## 6 24.052 23.726 23.797 23.451 23.252 23.359 23.532 99.000 23.782
##          Stellar_Flag
## 1          0.50
## 2          0.50
## 3          0.50
## 4          1.00
## 5          0.76
## 6          0.50

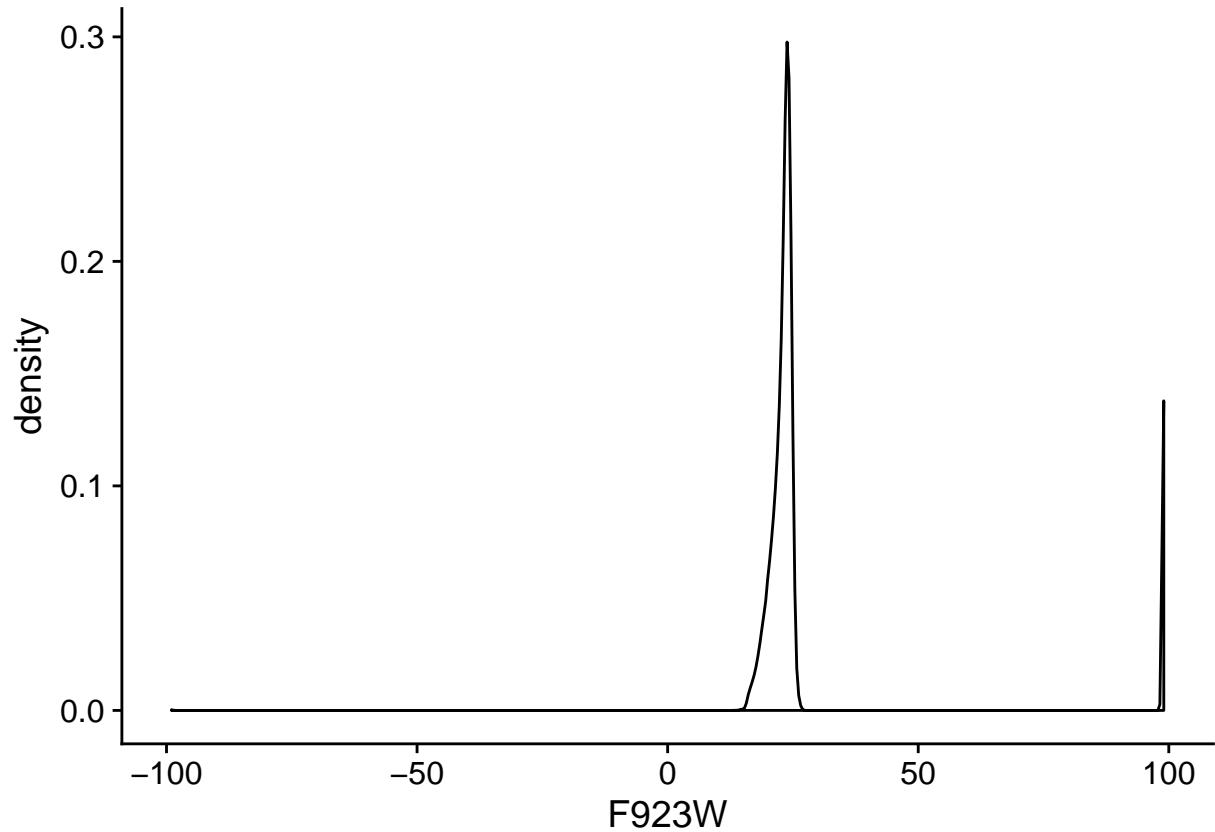
```

Revisión de los datos:

```

summary(util_data)
#ejemplo de funcion de densidad de una variable
ggplot(util_data, aes(x=F923W,xmin=20,xmax=30)) + geom_density()

```



```
##      objID          RA          DEC          stell
##  Min.   :8.142e+10  Min.   :36.66  Min.   : 0.5372  Min.   :0.0000
##  1st Qu.:8.143e+10  1st Qu.:138.94  1st Qu.: 2.4947  1st Qu.:0.0900
##  Median :8.145e+10  Median :189.61  Median :46.0130  Median :0.4600
##  Mean   :8.145e+10  Mean   :191.16  Mean   :33.1206  Mean   :0.4228
##  3rd Qu.:8.147e+10  3rd Qu.:243.19  3rd Qu.:54.0654  3rd Qu.:0.6500
##  Max.   :8.148e+10  Max.   :356.96  Max.   :62.4312  Max.   :1.0000
##      F365W          F396W          F427W          F458W
##  Min.   :-99.00      Min.   :-99.00      Min.   :-99.00      Min.   :-99.00
##  1st Qu.: 23.98      1st Qu.: 23.80      1st Qu.: 23.57      1st Qu.: 23.32
##  Median : 25.18      Median : 25.10      Median : 24.94      Median : 24.77
##  Mean   : 46.23      Mean   : 37.87      Mean   : 34.21      Mean   : 30.90
##  3rd Qu.: 99.00      3rd Qu.: 26.19      3rd Qu.: 25.93      3rd Qu.: 25.68
##  Max.   : 99.00      Max.   : 99.00      Max.   : 99.00      Max.   : 99.00
##      F489W          F520W          F551W          F582W
##  Min.   :-99.00      Min.   :-99.00      Min.   :-99.00      Min.   :-99.00
##  1st Qu.: 23.11      1st Qu.: 22.90      1st Qu.: 22.73      1st Qu.: 22.57
##  Median : 24.60      Median : 24.44      Median : 24.26      Median : 24.14
##  Mean   : 27.23      Mean   : 26.22      Mean   : 27.02      Mean   : 26.08
##  3rd Qu.: 25.47      3rd Qu.: 25.32      3rd Qu.: 25.15      3rd Qu.: 25.03
##  Max.   : 99.00      Max.   : 99.00      Max.   : 99.00      Max.   : 99.00
##      F613W          F644W          F675W          F706W
##  Min.   :-99.00      Min.   :-99.00      Min.   :-99.00      Min.   :-99.00
##  1st Qu.: 22.48      1st Qu.: 22.38      1st Qu.: 22.32      1st Qu.: 22.25
##  Median : 24.07      Median : 24.01      Median : 23.96      Median : 23.90
##  Mean   : 24.21      Mean   : 23.92      Mean   : 23.64      Mean   : 23.43
```

```

## 3rd Qu.: 24.95   3rd Qu.: 24.90   3rd Qu.: 24.85   3rd Qu.: 24.78
## Max.    : 99.00   Max.    : 99.00   Max.    : 99.00   Max.    : 99.00
##          F737W      F768W      F799W      F830W
## Min.    :-99.00   Min.    :-99.00   Min.    :-99.00   Min.    :-99.00
## 1st Qu.: 22.19   1st Qu.: 22.13   1st Qu.: 22.10   1st Qu.: 22.05
## Median  : 23.84   Median  : 23.79   Median  : 23.75   Median  : 23.71
## Mean    : 23.64   Mean    : 23.46   Mean    : 23.59   Mean    : 24.46
## 3rd Qu.: 24.71   3rd Qu.: 24.64   3rd Qu.: 24.62   3rd Qu.: 24.59
## Max.    : 99.00   Max.    : 99.00   Max.    : 99.00   Max.    : 99.00
##          F861W      F892W      F923W      F954W
## Min.    :-99.00   Min.    :12.70   Min.    :-99.00   Min.    :12.76
## 1st Qu.: 22.00   1st Qu.:21.95   1st Qu.: 21.93   1st Qu.:21.86
## Median  : 23.65   Median  :23.53   Median  : 23.48   Median  :23.15
## Mean    : 24.16   Mean    :27.11   Mean    : 30.28   Mean    :32.88
## 3rd Qu.: 24.54   3rd Qu.:24.41   3rd Qu.: 24.38   3rd Qu.:24.04
## Max.    : 99.00   Max.    :99.00   Max.    : 99.00   Max.    :99.00
##          J         H         KS        F814W
## Min.    :-99.00   Min.    :-99.00   Min.    :-99.00   Min.    :13.74
## 1st Qu.: 21.73   1st Qu.: 21.58   1st Qu.: 21.60   1st Qu.:22.04
## Median  : 23.45   Median  : 23.24   Median  : 23.23   Median  :23.65
## Mean    : 27.58   Mean    : 31.14   Mean    : 37.72   Mean    :23.04
## 3rd Qu.: 24.50   3rd Qu.: 24.37   3rd Qu.: 24.70   3rd Qu.:24.47
## Max.    : 99.00   Max.    : 99.00   Max.    : 99.00   Max.    :26.51
## Stellar_Flag
## Min.    :0.0000
## 1st Qu.:0.5000
## Median :0.5000
## Mean   :0.4459
## 3rd Qu.:0.5000
## Max.   :1.0000

```

### 3.1 Limpieza adicional

Debido a los valores no observados o no detectados (99 y -99) los valores estadísticos que nos proporciona R para cada variable no son reales. Eliminamos estos datos no válidos (-99) y los convertimos en NA. Si miramos la distribución de densidad de los datos y sabiendo la limitación en sensibilidad de las cámaras que los tomaron, además de los campos señalados como no detectados (99), los datos más allá de valores en torno a 25-26 magnitudes son poco fiables. Podríamos eliminar estos datos o como hemos optado en este caso, sustituir los valores por encima del límite por el propio límite. Para cada magnitud buscamos el límite en la distribución de densidad y hacemos la sustitución, incluyendo los valores no detectados (99).

Tras examinar cada variable y determinar el pico máximo de la función de densidad, dejamos cada filtro con su límite.

```

#Convertir -99 en NA
util_data[c(4:27)][(util_data[,c(4:27)] == -99)] <- NA
#Asignacion de limites
util_data$F365W[(util_data$F365W == 99) | (util_data$F365W > 25.2)] <- 25.2
util_data$F396W[(util_data$F396W == 99) | (util_data$F396W > 25.2)] <- 25.2
util_data$F427W[(util_data$F427W == 99) | (util_data$F427W > 25.2)] <- 25.2
util_data$F458W[(util_data$F458W == 99) | (util_data$F458W > 25.2)] <- 25.2
util_data$F489W[(util_data$F489W == 99) | (util_data$F489W > 25.2)] <- 25.2
util_data$F520W[(util_data$F520W == 99) | (util_data$F520W > 25.0)] <- 25.0
util_data$F551W[(util_data$F551W == 99) | (util_data$F551W > 24.9)] <- 24.9

```

```

util_data$F582W[(util_data$F582W == 99) | (util_data$F582W > 24.8)] <- 24.8
util_data$F613W[(util_data$F613W == 99) | (util_data$F613W > 24.8)] <- 24.8
util_data$F644W[(util_data$F644W == 99) | (util_data$F644W > 24.7)] <- 24.7
util_data$F675W[(util_data$F675W == 99) | (util_data$F675W > 24.7)] <- 24.7
util_data$F706W[(util_data$F706W == 99) | (util_data$F706W > 24.7)] <- 24.7
util_data$F737W[(util_data$F737W == 99) | (util_data$F737W > 24.6)] <- 24.6
util_data$F768W[(util_data$F768W == 99) | (util_data$F768W > 24.5)] <- 24.5
util_data$F799W[(util_data$F799W == 99) | (util_data$F799W > 24.5)] <- 24.5
util_data$F830W[(util_data$F830W == 99) | (util_data$F830W > 24.3)] <- 24.3
util_data$F861W[(util_data$F861W == 99) | (util_data$F861W > 24.3)] <- 24.3
util_data$F892W[(util_data$F892W == 99) | (util_data$F892W > 24.1)] <- 24.1
util_data$F923W[(util_data$F923W == 99) | (util_data$F923W > 23.9)] <- 23.9
util_data$F954W[(util_data$F954W == 99) | (util_data$F954W > 23.4)] <- 23.4
util_data$J[(util_data$J == 99) | (util_data$J > 24.0)] <- 24.0
util_data$H[(util_data$H == 99) | (util_data$H > 23.6)] <- 23.6
util_data$KS[(util_data$KS == 99) | (util_data$KS > 23.4)] <- 23.4
util_data$F814W[(util_data$F814W == 99) | (util_data$F814W > 24.4)] <- 24.4

head(util_data)

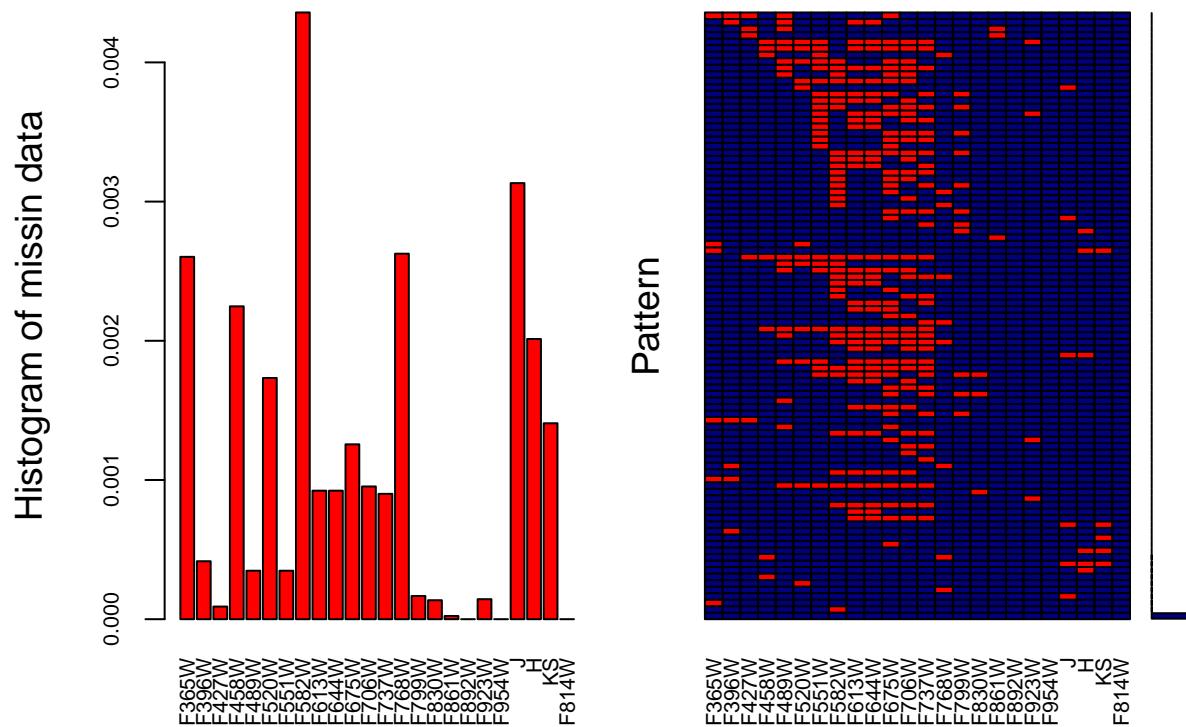
##          objID      RA      DEC stell  F365W  F396W  F427W  F458W  F489W
## 1 81481409774 356.9394 15.3444  0.64 25.200 25.144 25.009 24.800 23.423
## 2 81481409790 356.9355 15.3440  0.40 25.200 25.200 25.200 25.200 25.200
## 3 81481409773 356.9335 15.3442  0.45 25.200 24.630 25.200 25.200 25.200
## 4 81481409588 356.9476 15.3481  0.99 19.165 18.910 18.657 18.377 18.252
## 5 81481409853 356.9159 15.3424  0.91 25.200 25.200 25.200 25.200 24.965
## 6 81481409764 356.9278 15.3448  0.36 25.200 25.200 25.200 25.200 25.200
##      F520W  F551W  F582W  F613W  F644W  F675W  F706W  F737W  F768W  F799W
## 1 24.081 24.125 24.477 24.255 24.204 23.824 23.884 23.990 24.260 23.526
## 2 25.000 24.900 24.800 24.800 24.700 24.700 24.700 24.600 24.500 24.500
## 3 25.000 24.805 24.800 24.800 24.700 24.700 24.700 24.600 24.500 24.500
## 4 18.138 18.211 18.117 18.106 17.994 17.985 17.862 17.752 17.786 17.804
## 5 24.862 24.442 23.908 23.570 23.510 23.547 23.170 22.922 22.826 22.580
## 6 24.530 24.771 24.800 24.628 24.559 24.406 24.442 24.486 23.884 23.943
##      F830W  F861W  F892W  F923W  F954W      J      H      KS  F814W
## 1 23.554 23.972 23.575 23.178 22.678 23.971 23.600 23.400 23.498
## 2 24.300 24.300 24.100 23.900 23.400 24.000 23.600 23.400 24.400
## 3 24.300 24.300 24.100 23.900 23.400 24.000 23.600 23.400 24.400
## 4 17.671 17.763 17.713 17.626 17.804 17.770 17.959 18.328 17.609
## 5 22.675 22.439 22.230 22.081 22.230 21.992 21.782 22.274 22.473
## 6 24.052 23.726 23.797 23.451 23.252 23.359 23.532 23.400 23.782
##      Stellar_Flag
## 1          0.50
## 2          0.50
## 3          0.50
## 4          1.00
## 5          0.76
## 6          0.50

```

## 4 Revisión de los datos faltantes

```
#Contar los NAs
not_NA <- na.omit(util_data)
dim.data.frame(util_data) [1]
dim.data.frame(util_data) [1]-dim.data.frame(not_NA) [1]
dim.data.frame(not_NA) [1]

# Visualizacion de valores faltantes
aggr_plot <- util_data %>% select(F365W:F814W) %>% aggr(col=c('navyblue','red'),
numbers=TRUE, cex.axis=.7, gap=3,
ylab=c("Histogram of missin data","Pattern"))
```



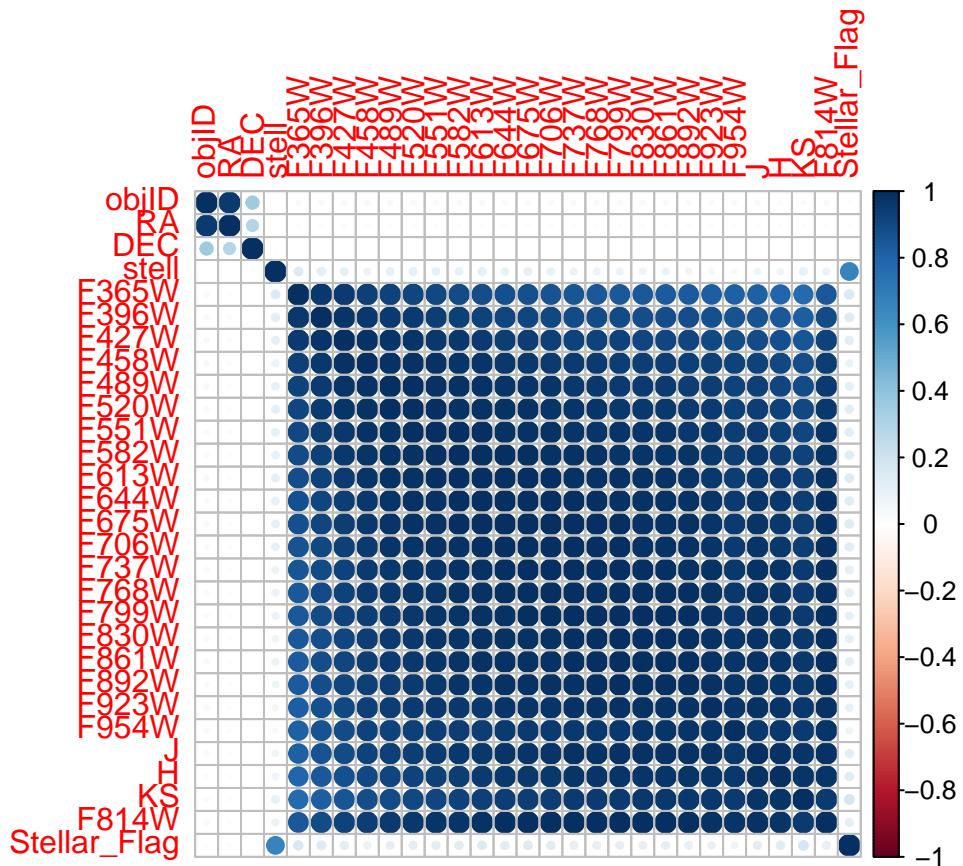
```
## [1] 132154
## [1] 2470
## [1] 129684
```

Aunque pueda parecer que hay alguna tendencia, los datos faltantes se refieren a aquellos datos no observados, principalmente por motivos técnicos o de mal tiempo, con lo que deberían ser aleatorios.

## 5 Exploración de datos: revisión de dependencias entre variables

```
#Eliminamos los datos faltantes (NA) para realizar las operaciones de analisis
util_data=na.omit(util_data)
```

```
corrplot(cor(util_data), method = "circle")
```



Como era de esperar, hay correlación entre las variables de flujo entre sí y entre las variables que señalan la naturaleza estelar de los objetos (stell y Stellar\_Flag). A priori no hay relación entre estas dos variables y las variables fotométricas, una razón puede ser que a partir de magnitud 21-22 se asignaba valor 0.5 a estas dos etiquetas, por la poca fiabilidad de los datos (objetos mas débiles con menor señal-ruido).

Ejemplo de matriz de correlación para 4 variables de ejemplo.

```
#scatterplotMatrix(~ F644W + F923W + J + H, data=na.omit(util_data), span=0.6)
```

Se aprecian desviaciones de la linearidad, la mayoría probablemente datos erróneos pero algunas sub-tendencias pueden ser significativas y pertenecer a subgrupos de poblaciones. Por esto, en principio no hacemos imputación de datos, ya que puede afectar al objetivo del trabajo.

Revisión si hay relación entre las etiquetas de estelaridad con todas las variables fotométricas.

```
lm.fit = lm(stell~F365W+F396W+F427W+F458W+F489W+F520W+F551W+F582W+F613W+
            F644W+F675W+F706W+F737W+F768W+F799W+F830W+F861W+F892W+
            F923W+J+H+KS+F814W, data=util_data)
summary (lm.fit)
lm.fit = lm(Stellar_Flag~F365W+F396W+F427W+F458W+F489W+F520W+F551W+F582W+
            F613W+F644W+F675W+F706W+F737W+F768W+F799W+F830W+F861W+
            F892W+F923W+J+H+KS+F814W, data=util_data)
summary (lm.fit)

##
```

```

## lm(formula = stell ~ F365W + F396W + F427W + F458W + F489W +
##     F520W + F551W + F582W + F613W + F644W + F675W + F706W + F737W +
##     F768W + F799W + F830W + F861W + F892W + F923W + J + H + KS +
##     F814W, data = util_data)
##
## Residuals:
##      Min      1Q Median      3Q      Max
## -2.74181 -0.22725 -0.00697  0.19212  2.18280
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) -0.677173  0.018271 -37.063 < 2e-16 ***
## F365W        0.079963  0.002612  30.616 < 2e-16 ***
## F396W        0.057114  0.003477  16.427 < 2e-16 ***
## F427W       -0.034096  0.003789 -8.998 < 2e-16 ***
## F458W       -0.230206  0.003782 -60.865 < 2e-16 ***
## F489W       -0.015072  0.004578 -3.292 0.000994 ***
## F520W        0.161311  0.004808  33.547 < 2e-16 ***
## F551W       -0.013086  0.004672 -2.801 0.005093 **  
## F582W       -0.031482  0.004963 -6.344 2.25e-10 ***
## F613W        0.135710  0.005645  24.040 < 2e-16 ***
## F644W       -0.049812  0.006082 -8.191 2.62e-16 ***
## F675W        0.138606  0.006015  23.045 < 2e-16 ***
## F706W        0.078293  0.006233  12.560 < 2e-16 ***
## F737W       -0.089774  0.005527 -16.243 < 2e-16 ***
## F768W       -0.021902  0.006250 -3.504 0.000458 *** 
## F799W       -0.057066  0.006081 -9.385 < 2e-16 ***
## F830W       -0.132112  0.005599 -23.598 < 2e-16 ***
## F861W        0.004363  0.005491  0.795 0.426844  
## F892W       -0.088701  0.004239 -20.926 < 2e-16 ***
## F923W       -0.188170  0.003710 -50.722 < 2e-16 ***
## J           -0.035435  0.003958 -8.952 < 2e-16 ***
## H           -0.009152  0.003285 -2.786 0.005343 **  
## KS          0.161253  0.002411  66.893 < 2e-16 ***
## F814W       0.225315  0.007257  31.046 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2862 on 129660 degrees of freedom
## Multiple R-squared:  0.1675, Adjusted R-squared:  0.1673 
## F-statistic:  1134 on 23 and 129660 DF, p-value: < 2.2e-16
##
## Call:
## lm(formula = Stellar_Flag ~ F365W + F396W + F427W + F458W + F489W +
##     F520W + F551W + F582W + F613W + F644W + F675W + F706W + F737W +
##     F768W + F799W + F830W + F861W + F892W + F923W + J + H + KS +
##     F814W, data = util_data)
##
## Residuals:
##      Min      1Q Median      3Q      Max
## -3.05210 -0.08123  0.01965  0.08256  1.64377
##
## Coefficients:

```

```

##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.783925  0.013419 -58.417 < 2e-16 ***
## F365W        0.083460  0.001918  43.507 < 2e-16 ***
## F396W        0.056654  0.002554  22.185 < 2e-16 ***
## F427W       -0.013738  0.002783 -4.936 7.98e-07 ***
## F458W        -0.196050  0.002778 -70.573 < 2e-16 ***
## F489W        -0.042090  0.003362 -12.519 < 2e-16 ***
## F520W         0.077683  0.003532  21.996 < 2e-16 ***
## F551W       -0.025679  0.003431 -7.484 7.28e-14 ***
## F582W       -0.005345  0.003645 -1.466 0.142557
## F613W        0.127671  0.004146  30.792 < 2e-16 ***
## F644W       -0.022522  0.004467 -5.042 4.61e-07 ***
## F675W        0.184419  0.004418  41.746 < 2e-16 ***
## F706W        0.038845  0.004578  8.485 < 2e-16 ***
## F737W       -0.109120  0.004059 -26.881 < 2e-16 ***
## F768W       -0.052985  0.004590 -11.542 < 2e-16 ***
## F799W       -0.074721  0.004466 -16.731 < 2e-16 ***
## F830W       -0.093906  0.004112 -22.837 < 2e-16 ***
## F861W        0.018140  0.004033 -4.498 6.86e-06 ***
## F892W       -0.087050  0.003113 -27.961 < 2e-16 ***
## F923W       -0.132921  0.002725 -48.781 < 2e-16 ***
## J            -0.010868  0.002907 -3.738 0.000186 ***
## H            0.021578  0.002413  8.942 < 2e-16 ***
## KS           0.177591  0.001771 100.302 < 2e-16 ***
## F814W        0.171826  0.005330  32.235 < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 0.2102 on 129660 degrees of freedom
## Multiple R-squared:  0.2552, Adjusted R-squared:  0.255
## F-statistic:  1931 on 23 and 129660 DF, p-value: < 2.2e-16

```

Los resultados indican una relación entre las magnitudes y las etiquetas (nivel de significación), aunque en algun caso no aparece, puede ser debido a la presencia de datos de mala calidad en dichas variables o al orden en que están metidos y sus correspondientes dependencias. Los residuos indican una simetría buena en ambas variables.

Correlación entre las etiquetas de estelaridad:

```

lm.fit =lm(Stellar_Flag~stell,data=util_data)
summary (lm.fit)

```

```

##
## Call:
## lm(formula = Stellar_Flag ~ stell, data = util_data)
##
## Residuals:
##      Min       1Q   Median       3Q      Max
## -0.74359 -0.13040  0.00851  0.15547  0.76575
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 0.2291002  0.0008494  269.7 <2e-16 ***
## stell       0.5144880  0.0016151  318.6 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

```

```

## 
## Residual standard error: 0.1824 on 129682 degrees of freedom
## Multiple R-squared:  0.439, Adjusted R-squared:  0.439
## F-statistic: 1.015e+05 on 1 and 129682 DF, p-value: < 2.2e-16

```

Como se ha mencionado antes, aunque debería haber una relación clara entre estas dos variables, reconocida en el nivel de significación de este ajuste, como hay un número de valores asignados a 0.5 por motivos de baja señal, la correlación no es tan grande como cabría esperar.

## 6 Reducción de variables

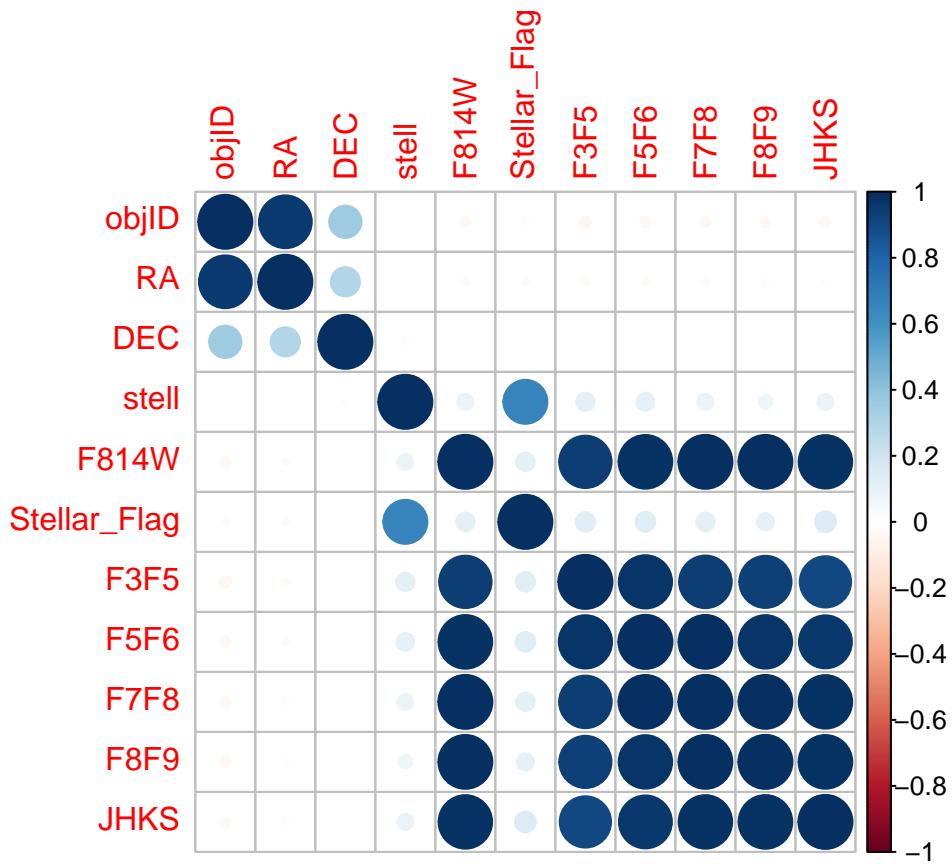
Hemos comprobado que hay relación y alta correlación entre las variables fotométricas. Para simplificar el estudio reducimos las 24 variables a 6, agrupadas de manera coherente, 4 grupos de filtros cercanos en longitud de onda y observados con la misma cámara, 1 grupo con los filtros infrarrojos (observados con la misma cámara) y otro con el filtro F814W que es posterior a los anteriores.

```

new_data <- util_data %>% mutate(F3F5=round(rowMeans(select(util_data,F365W:F520W)
,na.rm=TRUE),3))
new_data <- new_data %>% mutate(F5F6=round(rowMeans(select(util_data,F551W:F675W)
,na.rm=TRUE),3))
new_data <- new_data %>% mutate(F7F8=round(rowMeans(select(util_data,F706W:F830W)
,na.rm=TRUE),3))
new_data <- new_data %>% mutate(F8F9=round(rowMeans(select(util_data,F861W:F954W)
,na.rm=TRUE),3))
new_data <- new_data %>% mutate(JHKS=round(rowMeans(select(util_data,c(J,H,KS))
,na.rm=TRUE),3))
new_data <- new_data %>% select(-c(F365W:KS))

corrplot(cor(na.omit(new_data)), method = "circle")

```



## 7 Filtrado de Galaxias y Estrellas

Suponiendo fiable que los objetos cuyas etiquetas de estelaridad indican que son estrellas o galaxias con alta probabilidad, hemos formado unos ficheros separando estos objetos para realizar un entrenamiento de los datos e intentar determinar un modelo que discrimine entre ambos tipos.

Separación estrellas y galaxias y regresión logística con todas las variables. Añadimos una columna que asigna la probabilidad de galaxia (1) y no galaxia (0).

```
#FICHERO CON GALAXIAS
util_data_gal <- util_data %>% filter(stell < 0.1 & Stellar_Flag < 0.1) %>%
  mutate(galaxy = 1, prob = 1-stell) %>%
  select(objID, F365W:F814W,galaxy,prob)

#FICHERO CON ESTRELLAS
util_data_star <- util_data %>% filter(stell > 0.9 & Stellar_Flag > 0.9) %>%
  mutate(galaxy = 0, prob = stell) %>%
  select(objID, F365W:F814W,galaxy,prob)

#FICHERO CONJUNTO DE GALAXIAS Y ESTRELLAS
utilt<-rbind(util_data_gal,util_data_star)

#FICHERO DE OBJECTOS SIN CLASIFICACION DE ESTELARIDAD
util_data_unknow <- util_data %>% filter(stell > 0.1 & Stellar_Flag > 0.1
  & stell < 0.9 & Stellar_Flag < 0.9) %>%
  select(objID,F365W:F814W)
```

```

cat("num objetos que son galaxia:", dim.data.frame(util_data_gal)[1])
cat("\n num objetos que no son galaxia:", dim.data.frame(util_data_star)[1])
cat("\n num objetos que no sabemos lo que son:", dim.data.frame(util_data_unknown)[1])

## num objetos que son galaxia: 18360
## num objetos que no son galaxia: 9205
## num objetos que no sabemos lo que son: 77949

```

Aunque en los datos hay muchos más objetos que son galaxias (2 terceras partes), consideramos que no llegan a ser datos desbalanceados.

## 8 Análisis con clusterización

Además del estudio anterior, experimentamos con algoritmos de agrupación/clusterización. Queremos ver si de manera natural, objetos con similares características se agrupan, en este caso, aíñ sabiendo que entre los objetos estelares hay varias clases, queremos saber si podemos separar galaxias de estrellas.

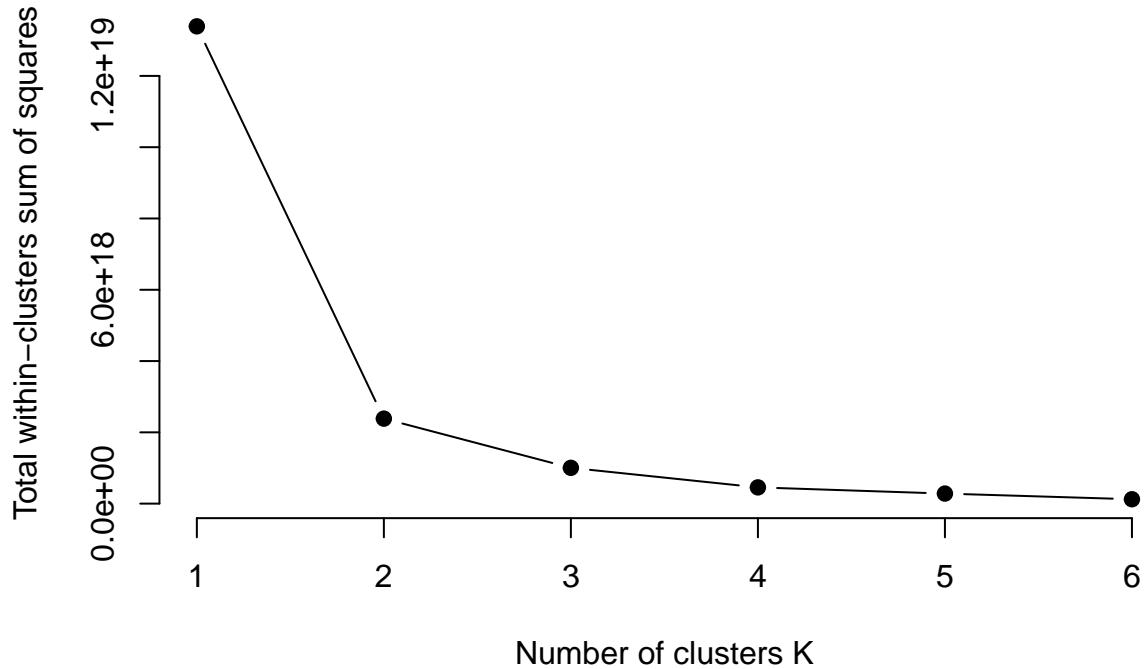
Inicialmente usamos el algoritmo Kmeans. Calculamos la curva de error para identificar cuál es el k que minimiza la suma del cuadrado de los errores manteniendo el mínimo k.

```

#calculo del error para valores de K hasta 6.
k.max <- 6
kdata <- na.omit(utilt)

wss <- sapply(1:k.max,
              function(k){kmeans(kdata, k, nstart=50,iter.max = 15 )$tot.withinss})
wss
plot(1:k.max, wss,
      type="b", pch = 19, frame = FALSE,
      xlab="Number of clusters K",
      ylab="Total within-clusters sum of squares")

```



```
## [1] 1.339050e+19 2.384776e+18 1.004135e+18 4.558714e+17 2.821213e+17
## [6] 1.233053e+17
```

El K que minimiza los errores esta entre 2 y 3.

Para K=2, calculamos la clasterizacion para los datos con reducción de variables que contienen solo los datos con clasificación conocida estrella o galaxia para comprobar el algoritmo.

```
kdata = utilt[,2:25]
km.out = kmeans(kdata, 2, nstart =200)
new_cluster <- kdata %>% mutate(grupo = km.out$cluster)
centroides = aggregate(kdata,by=list(km.out$cluster),FUN=mean)
t(centroides)

##           [,1]      [,2]
## Group.1  1.00000  2.00000
## F365W   21.52680 23.49156
## F396W   21.08753 23.27532
## F427W   20.63321 23.00898
## F458W   20.12493 22.71939
## F489W   19.92000 22.49868
## F520W   19.68794 22.27856
## F551W   19.47438 22.08190
## F582W   19.34495 21.93703
## F613W   19.24033 21.82198
## F644W   19.04943 21.69965
## F675W   19.00725 21.63327
## F706W   18.86890 21.55353
```

```

## F737W 18.70510 21.47239
## F768W 18.64054 21.41764
## F799W 18.56310 21.35247
## F830W 18.48673 21.29554
## F861W 18.45657 21.24826
## F892W 18.39982 21.20044
## F923W 18.34406 21.17116
## F954W 18.41903 21.17563
## J      18.14070 20.97690
## H      18.01879 20.81408
## KS     18.19348 20.85102
## F814W 18.53216 21.28985

```

Ahora calculamos el numero de elementos que hay en cada grupo enfrentado contra si es estrella ( 0 ) o galaxia ( 1 ).

```

tkmeans<-table(km.out$cluster,utilt$galaxy )
tkmeans

```

```

##
##          0      1
##    1  5101  3938
##    2  4104 14422

```

La tabla nos dice que ha clasificado 4104 en el grupo 1 siendo del grupo 0 (estrellas) y 14422 siendo 1 (galaxias). Equivalentemente en el grupo 2. Aunque a priori no sabemos qué grupo es cada cosa, no cabe duda de que el grupo 1 es el grupo de las galaxias por su elevado índice de acierto.

Calculamos el True Positive Rate (recall) y el False Positive Rate (ratio de falsa alarma) para visualizar la precisión del modelo y compararlo con los demás modelos que se van a estudiar a continuación.

Los que se clasifican 1 (galaxias) siendo galaxias (1) son 14422. Los que se clasifican 2 (estrellas) siendo estrellas (0) son 5101.

**NOTA:** Puede que distintas ejecuciones hagan que el grupo 1 se intercambie con el 2. Para asegurar un valor fijo, como hemos introducido una semilla al principio, fijaremos la tabla con los valores tal cual lo hemos definido antes.

```

tkmeans = c(4104, 5101, 14422, 3938)
total_positive <- tkmeans[3]+tkmeans[4]
TP_kmean <- tkmeans[3]/total_positive
total_negative <- tkmeans[1]+tkmeans[2]
FP_kmean <- tkmeans[1]/total_negative
FP_kmean
TP_kmean

```

```

## [1] 0.4458446
## [1] 0.785512

```

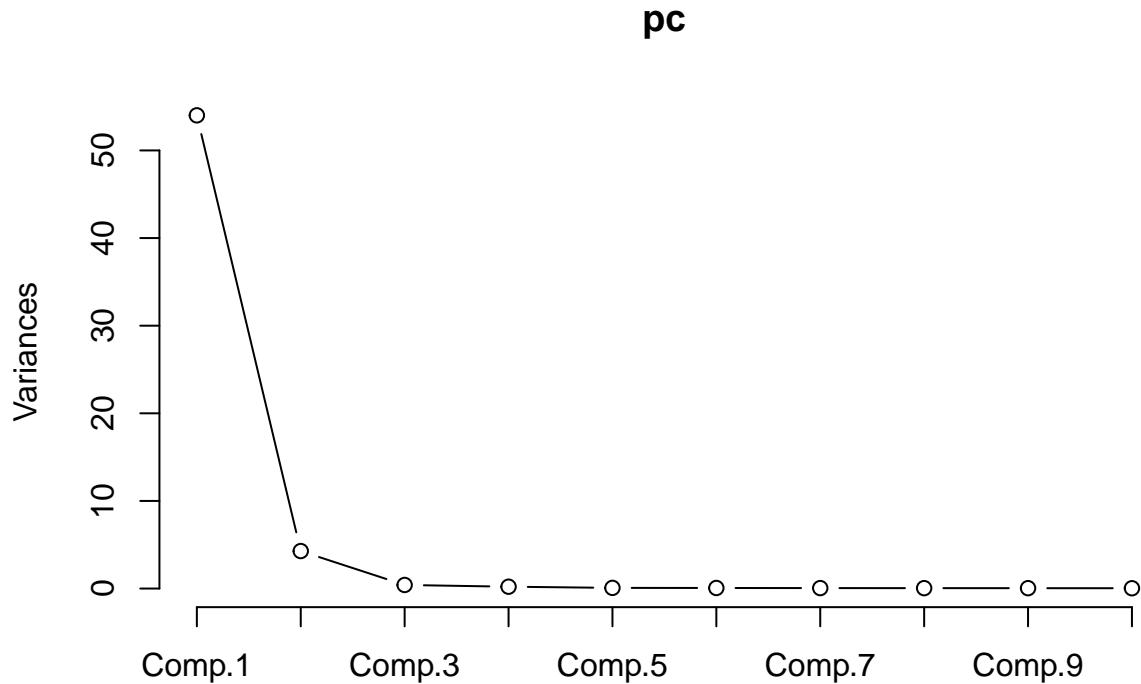
Podemos aplicar un algoritmo de componentes principales para reducir todas las variables a 2 o 3 dimensiones y así poder valorar si existen grupos marcados.

Para ello es interesante observar la pérdida de información en la reducción de variables.

```

pc <- princomp(kdata)
plot(pc, type='1')

```



```

summary(pc)
comp <- pc$scores[,1:3]

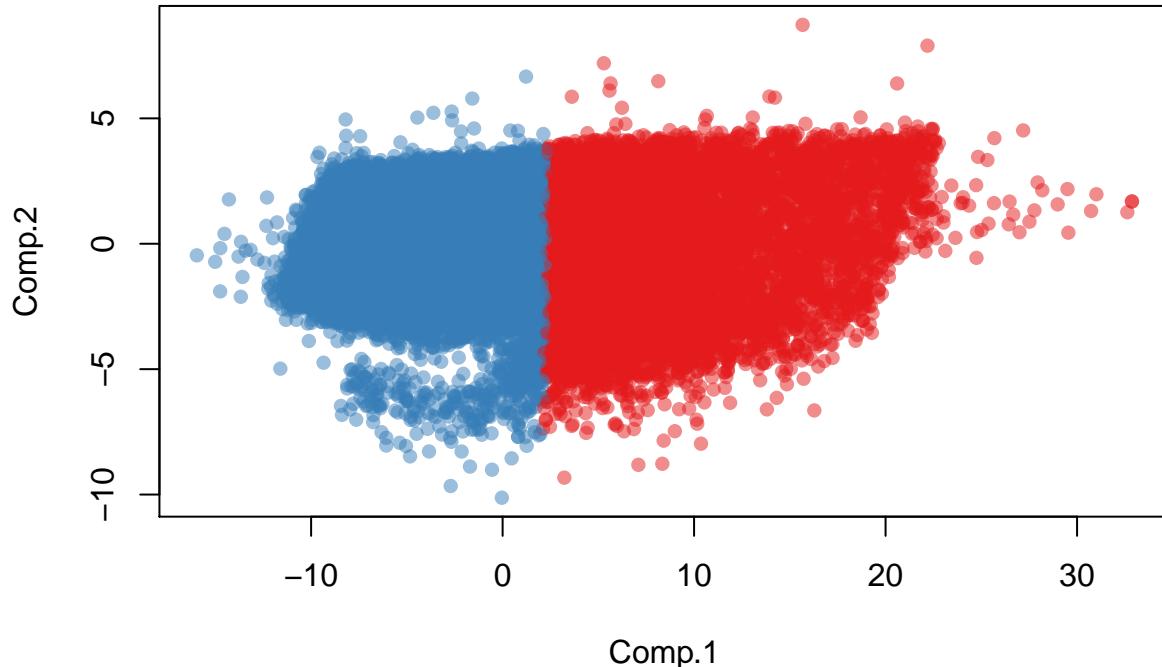
## Importance of components:
##                               Comp.1      Comp.2      Comp.3      Comp.4
## Standard deviation    7.3487828 2.06877433 0.638447859 0.461010965
## Proportion of Variance 0.9091086 0.07204622 0.006861765 0.003577729
## Cumulative Proportion  0.9091086 0.98115482 0.988016588 0.991594317
##                               Comp.5      Comp.6      Comp.7      Comp.8
## Standard deviation    0.2751874 0.254267552 0.2330345587 0.2159449910
## Proportion of Variance 0.0012748 0.001088346 0.0009141672 0.0007850028
## Cumulative Proportion  0.9928691 0.993957463 0.9948716301 0.9956566329
##                               Comp.9      Comp.10     Comp.11     Comp.12
## Standard deviation    0.2074154975 0.1886980971 0.1515093408 0.140525457
## Proportion of Variance 0.0007242147 0.0005994045 0.0003864237 0.000332426
## Cumulative Proportion  0.9963808476 0.9969802521 0.9973666758 0.997699102
##                               Comp.13     Comp.14     Comp.15     Comp.16
## Standard deviation    0.132294891 0.1278773175 0.1210806059 0.1194159173
## Proportion of Variance 0.000294626 0.0002752783 0.0002467937 0.0002400542
## Cumulative Proportion  0.997993728 0.9982690061 0.9985157999 0.9987558541
##                               Comp.17     Comp.18     Comp.19     Comp.20
## Standard deviation    0.1132594540 0.1092269456 0.1043261944 0.1000079760
## Proportion of Variance 0.0002159404 0.0002008374 0.0001832195 0.0001683659
## Cumulative Proportion  0.9989717945 0.9991726319 0.9993558514 0.9995242173
##                               Comp.21     Comp.22     Comp.23     Comp.24
## Standard deviation    0.0890944137 0.0866506460 0.0819797885 0.0780802111

```

```
## Proportion of Variance 0.0001336244 0.0001263946 0.0001131354 0.0001026282
## Cumulative Proportion 0.9996578417 0.9997842363 0.9998973718 1.0000000000
```

Se puede observar que la reducción a dos y tres componentes no conlleva una gran pérdida de información. Por tanto podemos aplicar Kmeans al set de datos reducido a dos variables y graficamos dos componentes.

```
k <- kmeans(comp, 2, nstart=25, iter.max=1000)
library(RColorBrewer)
library(scales)
palette(alpha(brewer.pal(9,'Set1')), 0.5)
plot(comp, col=k$clust, pch=16)
```



Graficamos la clusterización en 3 dimensiones para terminar de apreciar los grupos.

```
library(rgl)
#plot3d(comp[,1], comp[,3], comp[,2], col=k$clust)
```

Gracias al gráfico en 3D se puede observar que la tercera componente no contiene mucha relevancia, sin embargo se pueden apreciar dos grupos. La nube más densa corresponde con las galaxias. Dada la naturaleza del algoritmo existe una zona de confusión en el centro de la nube donde existe un corte bien marcado. Esto puede deberse al ruido del set de datos.

Podemos afirmar que la clusterización es un método a priori válido para obtener información de nuestro set de datos sin embargo K means no es el algoritmo ideal para este problema, debiendo optar por un algoritmo basado en densidades.

## 9 Regresión logistica

### 9.1 Ficheros de entrenamiento y testeo:

Lo primero de todo es separar los datos en ficheros de training-test, tomando un 70% como training y 30% como test para luego realizar contrastes.

```
n_data=dim(utilt)[1]
n_train=round(0.7*n_data)
n_test=n_data-n_train

indices=1:n_data
indices_train= sample(indices,n_train)
indices_test=indices[-indices_train]

train_data=utilt[indices_train,]
test_data=utilt[indices_test,]
dim(train_data)
dim(test_data)
class(train_data$galaxy)
head(train_data)

## [1] 19296    27
## [1] 8269    27
## [1] "numeric"
##           objID  F365W  F396W  F427W  F458W  F489W  F520W  F551W  F582W
## 459     81481401929 23.625 23.643 23.599 22.993 22.964 22.838 22.797 22.911
## 1462    81481107610 25.067 23.953 23.440 23.306 23.145 22.862 22.494 22.341
## 14897   81432302428 19.297 18.991 18.501 18.115 17.711 17.507 17.456 17.283
## 10208   81451107396 23.122 23.048 23.237 22.973 22.916 22.859 22.419 22.308
## 21409   81421107949 21.317 20.594 19.968 19.123 18.904 18.592 18.231 18.102
## 16885   81441203459 24.302 24.284 23.885 23.452 23.376 23.099 22.892 22.820
##           F613W  F644W  F675W  F706W  F737W  F768W  F799W  F830W  F861W
## 459      22.785 22.792 22.720 22.727 22.720 22.540 22.711 22.839 22.379
## 1462     22.199 21.985 21.943 21.848 21.734 21.736 21.606 21.591 21.535
## 14897   17.175 17.068 17.014 16.889 16.831 16.795 16.737 16.634 16.581
## 10208   22.151 22.143 22.195 22.068 21.805 22.189 22.092 22.068 22.029
## 21409   18.014 17.513 17.533 17.176 16.767 16.671 16.487 16.338 16.291
## 16885   22.848 22.761 22.683 22.743 22.548 22.546 22.602 22.460 22.341
##           F892W  F923W  F954W      J      H      KS  F814W galaxy prob
## 459      22.563 22.076 22.092 22.621 22.388 21.878 22.416      1 0.99
## 1462     21.466 21.247 21.486 21.117 20.863 20.964 21.522      1 0.96
## 14897   16.571 16.539 16.432 16.222 15.936 16.174 16.601      1 0.96
## 10208   22.082 22.076 22.698 21.764 21.788 22.128 22.056      1 0.97
## 21409   16.273 16.115 16.199 15.779 15.775 15.975 16.519      0 0.99
## 16885   22.466 22.319 22.551 22.027 21.825 21.882 22.496      1 0.97
```

Probamos el análisis de datos con todas las variables y con la lista de variables reducidas.

### 9.2 Regresión logistica con todas las variables usando el fichero de entrenamiento

```

glm_log=glm(formula = galaxy~F365W+F396W+F427W+F458W+F489W+F520W+F551W+F582W+
             F613W+F644W+F675W+F706W+F737W+F768W+F799W+
             F830W+F861W+F892W+F923W+J+H+KS+F814W,
             family = binomial, data = train_data)
summary(glm_log)

##
## Call:
## glm(formula = galaxy ~ F365W + F396W + F427W + F458W + F489W +
##       F520W + F551W + F582W + F613W + F644W + F675W + F706W + F737W +
##       F768W + F799W + F830W + F861W + F892W + F923W + J + H + KS +
##       F814W, family = binomial, data = train_data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -5.9279 -0.1592  0.0886  0.2308  8.4904
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.99620   0.46872 -6.392 1.63e-10 ***
## F365W       -1.16541   0.08825 -13.206 < 2e-16 ***
## F396W       -0.86619   0.12479 -6.941 3.89e-12 ***
## F427W       -0.67580   0.16076 -4.204 2.62e-05 ***
## F458W        2.39817   0.19241 12.464 < 2e-16 ***
## F489W        1.41110   0.27541  5.124 3.00e-07 ***
## F520W       -3.94123   0.27879 -14.137 < 2e-16 ***
## F551W        0.71542   0.27179   2.632  0.00848 **
## F582W        1.43122   0.22625   6.326 2.52e-10 ***
## F613W       -2.84117   0.32229  -8.816 < 2e-16 ***
## F644W        4.38145   0.42325  10.352 < 2e-16 ***
## F675W       -5.56193   0.39766 -13.987 < 2e-16 ***
## F706W        1.76636   0.40377   4.375 1.22e-05 ***
## F737W        5.65872   0.34486  16.409 < 2e-16 ***
## F768W        2.60602   0.45689   5.704 1.17e-08 ***
## F799W        5.07472   0.40266  12.603 < 2e-16 ***
## F830W        2.92900   0.35416   8.270 < 2e-16 ***
## F861W       -2.01143   0.37287  -5.394 6.87e-08 ***
## F892W       -1.64648   0.29430  -5.595 2.21e-08 ***
## F923W       -0.28736   0.23863  -1.204  0.22852
## J           3.02522   0.23233  13.021 < 2e-16 ***
## H          -1.49182   0.16474  -9.056 < 2e-16 ***
## KS          -4.05752   0.09939 -40.825 < 2e-16 ***
## F814W       -6.55107   0.44837 -14.611 < 2e-16 ***
##
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 24593.4 on 19295 degrees of freedom
## Residual deviance: 7108.9 on 19272 degrees of freedom
## AIC: 7156.9
##
## Number of Fisher Scoring iterations: 8

```

Observamos, al igual que cuando se hizo para el fichero original completo respecto a las variables etiquetas de estelaridad, que las variables estan relacionadas (nivel de significacion) salvo dos variables que segun el modelo no son relevantes. Probamos a quitarlas (F551W, F861W), pero podria ser solo por datos erroneos (aunque hemos limpiado bastante) o dependencias entre variables, el orden, etc.

### 9.3 Regresion logistica eliminando las dos variables:

```
glm_log_1=glm(formula = galaxy~F365W+F396W+F427W+F458W+F489W+F520W+F582W+
               F613W+F644W+F675W+F706W+F737W+F768W+F799W+F830W+
               F923W+F892W+J+H+KS+F814W,
               family = binomial, data = train_data)
summary(glm_log_1)

##
## Call:
## glm(formula = galaxy ~ F365W + F396W + F427W + F458W + F489W +
##       F520W + F582W + F613W + F644W + F675W + F706W + F737W + F768W +
##       F799W + F830W + F923W + F892W + J + H + KS + F814W, family = binomial,
##       data = train_data)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -5.7340 -0.1616  0.0905  0.2333  8.4904
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -2.90975   0.46612 -6.242 4.31e-10 ***
## F365W       -1.17402   0.08712 -13.477 < 2e-16 ***
## F396W       -0.84859   0.12310 -6.893 5.45e-12 ***
## F427W       -0.69699   0.15775 -4.418 9.94e-06 ***
## F458W        2.49445   0.18872 13.218 < 2e-16 ***
## F489W        1.56113   0.26998  5.782 7.36e-09 ***
## F520W       -3.75968   0.26610 -14.129 < 2e-16 ***
## F582W        1.65761   0.23226  7.137 9.54e-13 ***
## F613W       -2.73620   0.32738 -8.358 < 2e-16 ***
## F644W        4.17290   0.41439 10.070 < 2e-16 ***
## F675W       -5.33471   0.40461 -13.185 < 2e-16 ***
## F706W        1.53379   0.42202  3.634 0.000279 ***
## F737W        5.81936   0.34737 16.753 < 2e-16 ***
## F768W        2.22362   0.45123  4.928 8.31e-07 ***
## F799W        4.78489   0.44183 10.830 < 2e-16 ***
## F830W        2.42265   0.38192  6.343 2.25e-10 ***
## F923W       -0.45123   0.22977 -1.964 0.049551 *
## F892W       -1.94307   0.29371 -6.616 3.70e-11 ***
## J            2.96782   0.22560 13.155 < 2e-16 ***
## H           -1.51155   0.16706 -9.048 < 2e-16 ***
## KS           -4.07628   0.09912 -41.126 < 2e-16 ***
## F814W       -6.81092   0.44376 -15.348 < 2e-16 ***
##
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
```

```

##      Null deviance: 24593.4  on 19295  degrees of freedom
## Residual deviance:  7141.2  on 19274  degrees of freedom
## AIC: 7185.2
##
## Number of Fisher Scoring iterations: 8

```

Ambos modelos son equivalentes si miramos las variaciones y residuos y el factor AIC

Para visualizar la precision y sendibilidad del ajuste, pintamos la curva ROC de los modelos y creamos las tablas de eventos

```

prediction_train <- predict(glm_log, train_data, type = "response")
tglm <- table(train_data$galaxy, prediction_train > 0.5)
tglm

```

```

##
##      FALSE  TRUE
## 0  5933   520
## 1   450 12393

```

Predecimos para test:

```

prediction_test = predict(glm_log, test_data, type = "response")
tglm_test <- table(test_data$galaxy, prediction_test > 0.5)
tglm_test

```

```

##
##      FALSE  TRUE
## 0  2505   247
## 1   176 5341

```

Esto quiere decir que los que han salido TRUE son los que son mayores de 0.5, esto es, los que los clasificamos como 1 (galaxias). Inversamente para menor que 0.5. Por tanto, ha clasificado correctamente como galaxias 5337, y como estrellas 2507.

Calculamos el recall y el ratio de falsa alarma para los datos train puesto que el volumen de datos es semejante al utilizado en el kmeans.

```

total_pos <- tglm[2]+tglm[4]
TP_glm <- tglm[4]/total_pos
total_neg <- tglm[1]+tglm[3]
FP_glm <- tglm[3]/total_neg
FP_glm
TP_glm

```

```

## [1] 0.08058267
## [1] 0.9649615

```

```

z1_test = predict(glm_log, test_data, type = "response")

```

```

y_test <- test_data$galaxy

```

```

pred <- prediction(z1_test, y_test)

```

```

# Area bajo al curva de ROC

```

```

auc.tmp <- performance(pred,"auc");
auc_resume <- as.numeric(auc.tmp@y.values)

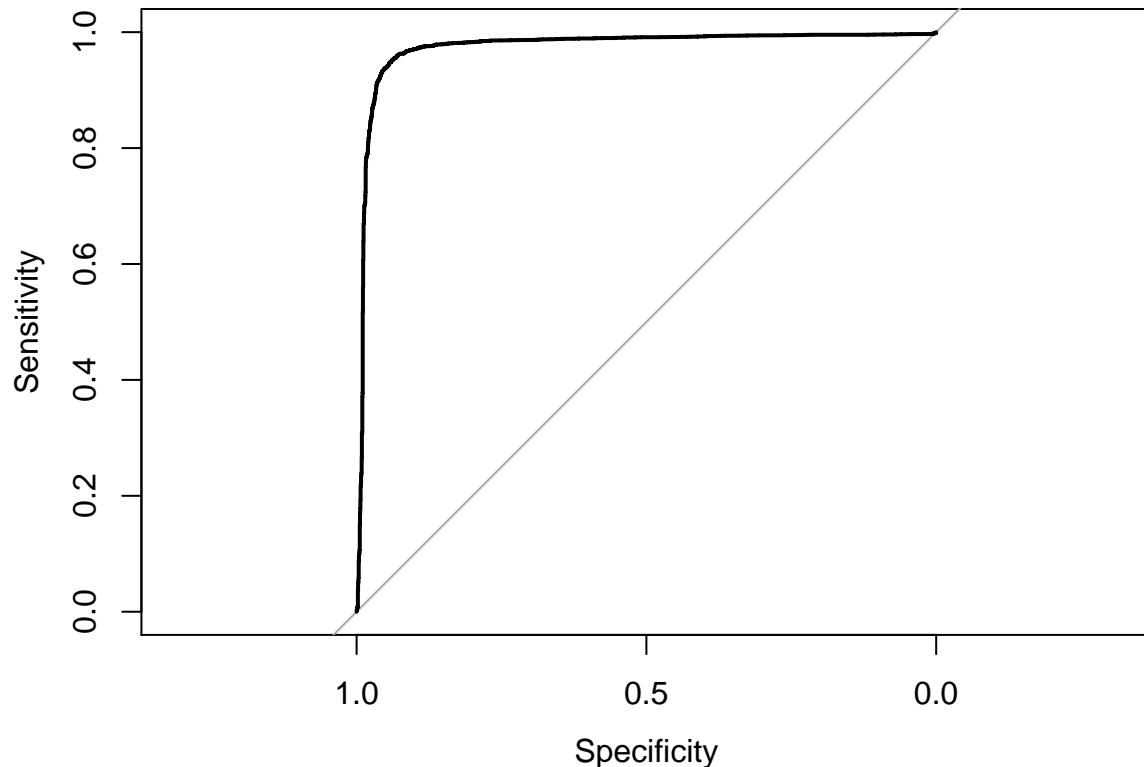
```

```

cat("El area bajo la curva ROC del modelo es de: ", auc_resume)

#Curva de ROC
g_test <- roc(galaxy ~ z1_test, data = test_data)
plot(g_test)

```



```
## El area bajo la curva ROC del modelo es de: 0.9746676
```

Obtenemos un alto grado de predicción. Además nos indica que el área bajo la curva ROC es cercano a 98.

Podemos utilizar la tabla ANOVA que en regresión tradicional para comparar el ajuste de estos dos modelos, ya que están anidados.

```
anova(glm_log, glm_log_1, test = "Chisq")
```

```

## Analysis of Deviance Table
##
## Model 1: galaxy ~ F365W + F396W + F427W + F458W + F489W + F520W + F551W +
##           F582W + F613W + F644W + F675W + F706W + F737W + F768W + F799W +
##           F830W + F861W + F892W + F923W + J + H + KS + F814W
## Model 2: galaxy ~ F365W + F396W + F427W + F458W + F489W + F520W + F582W +
##           F613W + F644W + F675W + F706W + F737W + F768W + F799W + F830W +
##           F923W + F892W + J + H + KS + F814W
##   Resid. Df Resid. Dev Df Deviance Pr(>Chi)
## 1      19272    7108.9
## 2      19274    7141.2 -2   -32.249 9.937e-08 ***

```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Como el valor del estadístico no es significativo, en principio, la eliminación de las variables puede considerarse válida.

## 9.4 Regresión logística para fichero con reducción de variables

Mismo análisis pero con el fichero que contiene la reducción de variables

```
#FICHERO CON GALAXIAS
new_data_gal <- new_data %>% filter(stell < 0.1 & Stellar_Flag < 0.1) %>%
  mutate(galaxy = 1, prob = 1-stell) %>%
  select(objID, F3F5:JHKS,F814W,galaxy,prob)

#FICHERO CON ESTRELLAS
new_data_star <- new_data %>% filter(stell > 0.9 & Stellar_Flag > 0.9) %>%
  mutate(galaxy = 0, prob = stell) %>%
  select(objID, F3F5:JHKS,F814W,galaxy,prob)

#FICHERO CONJUNTO DE GALAXIAS Y ESTRELLAS
newt<-rbind(new_data_gal,new_data_star)

#FICHERO DE OBJECTOS SIN CLASIFICACION DE ESTELARIDAD
new_data_unknown <- new_data %>% filter(stell > 0.1 & Stellar_Flag > 0.1
  & stell < 0.9 & Stellar_Flag < 0.9) %>% select(objID,F3F5:JHKS,F814W)

cat("num objetos que son galaxia:", dim.data.frame(new_data_gal)[1])
cat("\n num objetos que no son galaxia:", dim.data.frame(new_data_star)[1])
cat("\n num objetos que no sabemos lo que son:", dim.data.frame(new_data_unknown)[1])
```

```
## num objetos que son galaxia: 18360
## num objetos que no son galaxia: 9205
## num objetos que no sabemos lo que son: 77949
```

Separación de datos en ficheros de training-test, tomando un 70% como training y 30% como test.

```
n_new_data=dim(newt)[1]
n_new_train=round(0.7*n_new_data)
n_new_test=n_new_data-n_new_train

indices=1:n_new_data
indices_new_train= sample(indices,n_new_train)
indices_new_test=indices[-indices_new_train]

new_train_data=newt[indices_new_train,]
new_test_data=newt[indices_new_test,]
```

Regresión logística con todas las variables usando el fichero de entrenamiento

```
new_glm_log=glm(formula = galaxy~F3F5+F5F6+F7F8+F8F9+JHKS+F814W,
  family = binomial, data = new_train_data)
summary(new_glm_log)

##
## Call:
## glm(formula = galaxy ~ F3F5 + F5F6 + F7F8 + F8F9 + JHKS + F814W,
```

```

##      family = binomial, data = new_train_data)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -7.6258  -0.3074   0.1602   0.3604   8.4904
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) -5.66216   0.36588 -15.47 <2e-16 ***
## F3F5        -2.59720   0.07826 -33.19 <2e-16 ***
## F5F6        -5.23370   0.21147 -24.75 <2e-16 ***
## F7F8        24.58744   0.57580  42.70 <2e-16 ***
## F8F9        -3.42315   0.28913 -11.84 <2e-16 ***
## JHKS        -6.95113   0.10623 -65.44 <2e-16 ***
## F814W       -5.97019   0.35527 -16.80 <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 24636  on 19295  degrees of freedom
## Residual deviance: 10137  on 19289  degrees of freedom
## AIC: 10151
##
## Number of Fisher Scoring iterations: 7

```

Este modelo se puede considerar peor que los dos anteriores, si comparamos los valores de los residuos y del factor AIC, aunque no es muy grande.

Tabla de eventos

```

z_new = predict(new_glm_log, new_test_data, type = "response")
tglm_new = table(new_test_data$galaxy, z_new > 0.5)
tglm_new

```

```

##
##      FALSE TRUE
##      0    2441 280
##      1    184 5364

total_pos <- tglm_new[2]+tglm_new[4]
TP_glm_new <- tglm_new[4]/total_pos
total_neg <- tglm_new[1]+tglm_new[3]
FP_glm_new <- tglm_new[3]/total_neg
FP_glm_new
TP_glm_new

```

```

## [1] 0.1029033
## [1] 0.9668349

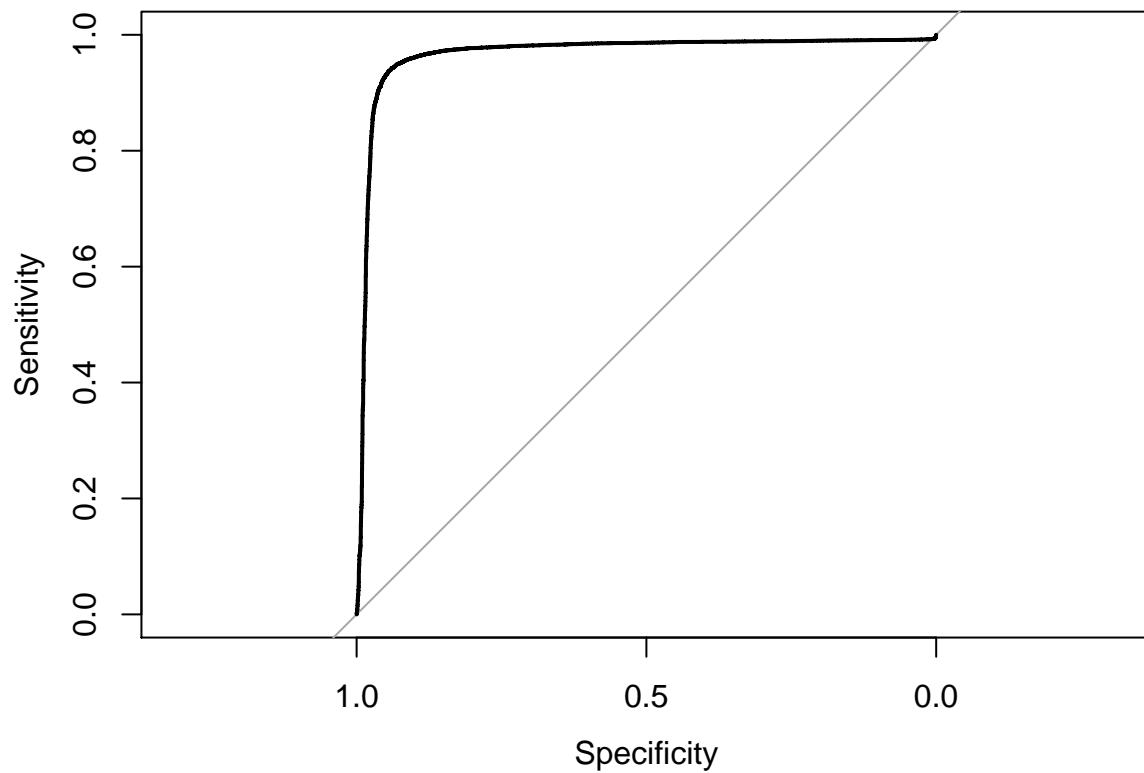
```

Curva ROC de los modelo

```

prob_new_glm = predict(new_glm_log, type = c("response"))
g_new <- roc(galaxy ~ prob_new_glm, data = new_train_data)
plot(g_new)

```



Cálculo de la precisión del modelo

```
# Calculamos sobre le fichero de entrenamiento:
y_new <- new_test_data$galaxy

new_pred <- prediction(z_new, y_new)

# Area bajo al curva de ROC
auc.tmp_new <- performance(new_pred, "auc");
auc_resume_new <- as.numeric(auc.tmp_new@y.values)

cat("El area bajo la curva ROC del nuevo modelo es de: ", auc_resume_new)

## El area bajo la curva ROC del nuevo modelo es de: 0.9698864
```

Como vemos es muy similar a los valores obtenidos con los otros dos modelos (~98%), pero algo inferior.

## 10 Predicción de datos

Con el modelo que da mejor resultado, asignamos un grupo (galaxia o no) a la lista de objetos de los cuales no sabemos su naturaleza.

```
#Prediccion para los datos desconocidos
pred <- predict(glm_log_1, util_data_unknown, type = "response")
head(pred)
```

```

probs <- exp(pred)/(1+exp(pred)) # Da la probabilidad de que y=1

#Anadimos predicción y la probabilidad asociada
util_data_unknown <- util_data_unknown %>% mutate( galaxy =
    trunc(pred+0.5)) %>% mutate(prob=probs)

final_data <- rbind(utilt,util_data_unknown)
head(final_data)

##          1         2         3         4         5         6
## 0.9997598 0.9999095 0.9999442 0.9999561 0.9998822 0.9841971
##      objID F365W F396W F427W F458W F489W F520W F551W F582W
## 1 81481409807 23.886 23.766 23.802 23.973 23.381 23.111 22.755 22.882
## 2 81481409487 22.629 22.024 21.879 21.786 21.286 21.000 20.882 20.552
## 3 81481409686 19.896 19.677 19.375 18.925 18.676 18.435 18.497 18.438
## 4 81481409323 23.442 23.663 23.506 23.390 23.510 23.105 22.909 22.815
## 5 81481409689 22.972 22.821 22.933 22.594 22.667 22.465 22.265 22.076
## 6 81481409701 23.337 23.125 23.133 23.111 23.094 22.758 22.445 22.569
##      F613W F644W F675W F706W F737W F768W F799W F830W F861W F892W
## 1 22.807 22.654 22.617 22.623 22.633 22.586 22.604 22.739 22.401 22.252
## 2 20.442 20.173 20.084 19.906 19.810 19.767 19.691 19.566 19.573 19.473
## 3 18.333 18.147 18.110 17.979 17.717 17.780 17.805 17.654 17.632 17.584
## 4 22.832 22.756 22.684 22.582 22.762 22.584 22.700 22.827 22.697 22.499
## 5 21.952 21.826 21.779 21.655 21.563 21.679 21.658 21.604 21.331 21.485
## 6 22.532 22.421 22.401 21.968 22.261 22.396 22.500 22.427 22.478 22.392
##      F923W F954W      J      H     KS F814W galaxy prob
## 1 22.677 22.603 22.262 22.179 22.477 22.405      1 0.97
## 2 19.011 19.447 19.000 18.687 18.382 19.479      1 0.97
## 3 17.508 17.619 17.171 17.036 16.936 17.567      1 0.97
## 4 21.900 22.342 22.378 22.218 22.846 22.450      1 0.97
## 5 21.378 21.253 21.141 21.186 20.762 21.398      1 0.98
## 6 22.419 21.733 22.262 21.793 21.745 22.166      1 0.98

```

La regresión logística parece funcionar bien, con aproximadamente un 0.98% de recuperación de los datos/test. Pero en la ejecución para datos desconocidos es menor, discriminando bien estrellas de tipos frios pero con mucha confusión entre galaxias y tipos de estrellas calientes. Esto lo sabemos porque se ha comparado los resultados con índices de color de la manera tradicional en astronomía mostrada en la figura siguiente.

```

final_datag <- final_data %>% filter(galaxy==1)
final_datas <- final_data %>% filter(galaxy==0)

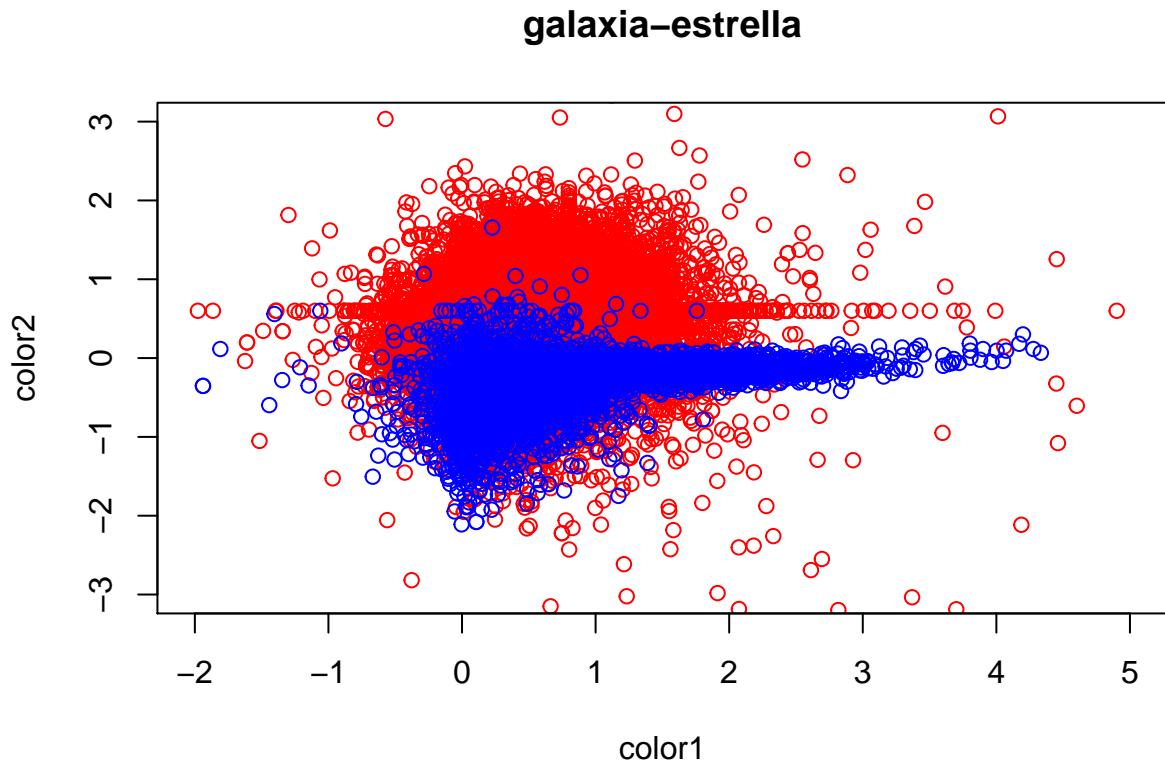
gg <- util_data_unknown %>% filter(galaxy==1)
ss <- util_data_unknown %>% filter(galaxy==0)

xeg=final_datag$F644W-final_datag$F923W
yeg=final_datag$J-final_datag$KS
xes=final_datas$F644W-final_datas$F923W
yes=final_datas$J-final_datas$KS

xgg= gg$F644W-gg$F923W
ygg= gg$J-gg$KS
xss= ss$F644W-ss$F923W
yss= ss$J-ss$KS

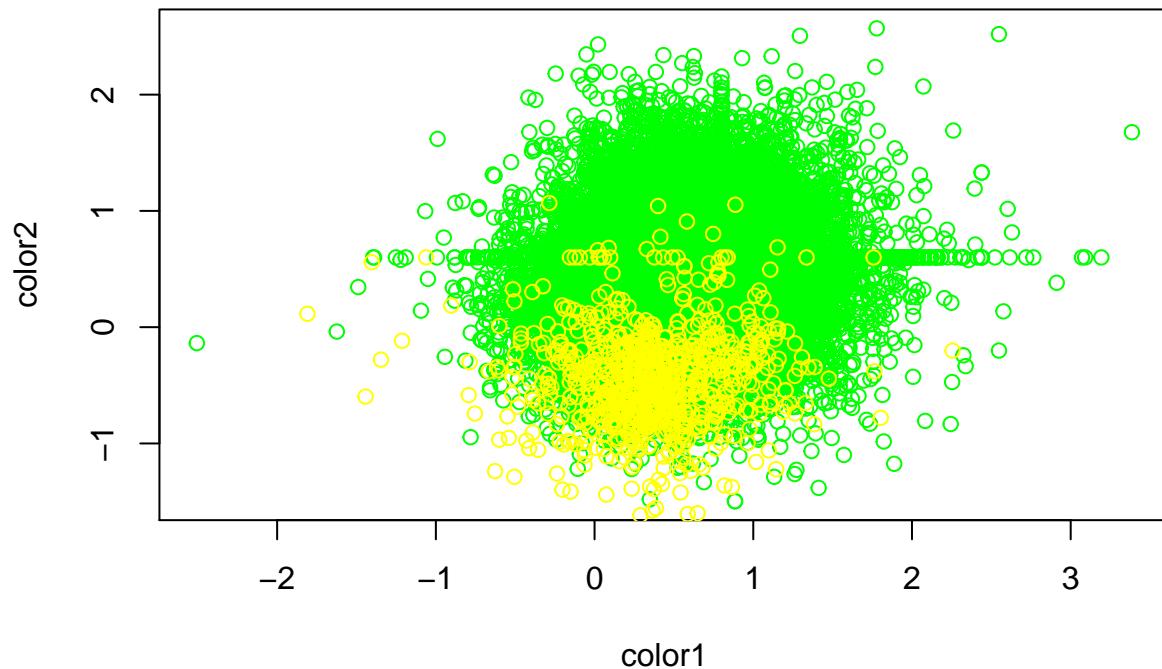
```

```
plot(xeg,yeg,col='red',xlim=c(-2,5),ylim=c(-3,3),main="galaxia-estrella",
      xlab="color1",ylab="color2")
points(xes,yes,col='blue')
```



```
plot(xgg,ygg,col='green',main="galaxia-estrella",xlab="color1",ylab="color2")
points(xss,yss,col='yellow')
```

## galaxia–estrella



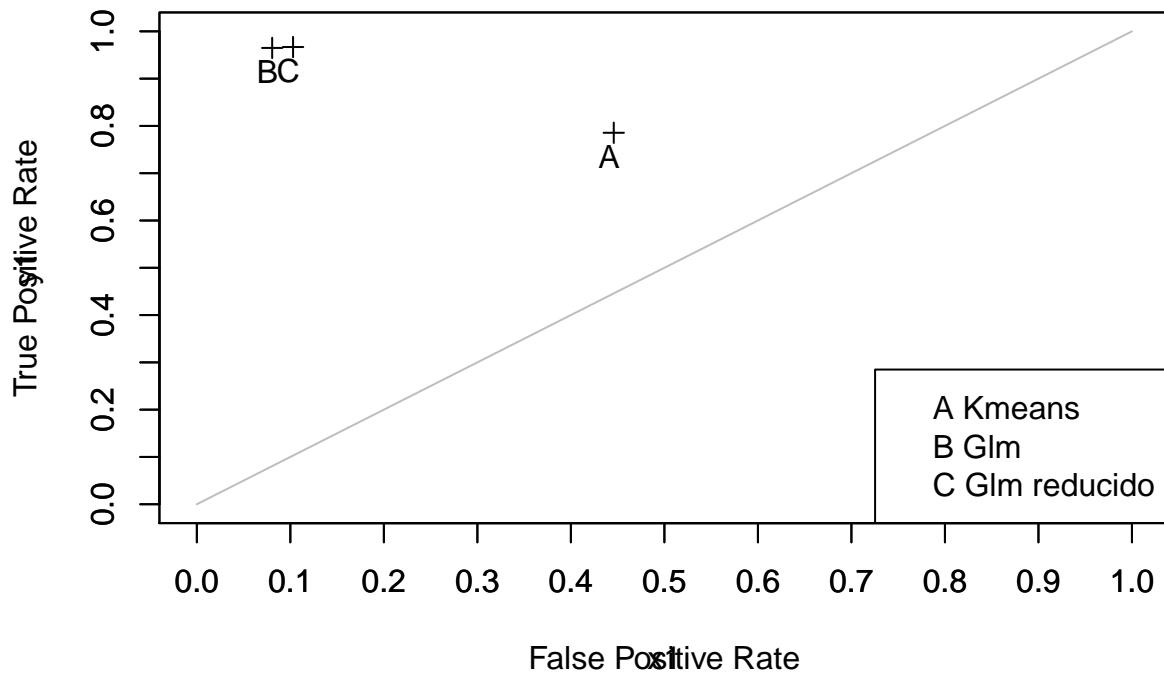
El rojo (y verde) señala galaxias y el azul (y amarillo) estrellas. Rojos y azules son la recuperación de datos y verde y amarillo los desconocidos, estando en la zona de confusión entre estrellas calientes y galaxias. Los índices astronómicos tampoco pueden discriminar mejor sin usar otras técnicas.

Una de las maneras posibles que nos planteamos para mejorar el modelo fue mejorar la limpieza inicial de los datos con el parámetro de señal ruido pero al realizar la limpieza para una señal (parámetro s2n del fichero original) por encima de cierto límite, la mejora no fue casi apreciable y por eso no lo hemos incluído.

## 11 Evaluación de modelos. Comparación

```
par(lab=c(x=8,y=8,len=1))
x <- c(FP_kmean, FP_glm, FP_glm_new)
y <- c(TP_kmean, TP_glm, TP_glm_new)
plot(x, y, pch=3, main="Evaluación de Modelos", xlab="False Positive Rate",
      ylab="True Positive Rate", xlim=c(0,1), ylim=c(0,1))
text(x-0.005,y-0.05,c("A", "B", "C"))
legend("bottomright",legend=c("A Kmeans", "B Glm", "C Glm reducido"))
par(new="True")
x1 <- seq(0,1,0.1)
y1 <- x1
plot(x1, y1, type="l", col="grey")
```

## Evaluación de Modelos



Como ya hemos ido viendo, parece que los modelos de regresión B y C son equivalentes y son mejores que el modelo de K-means en este caso.

## 12 Modelo con Máquinas de Vector Soporte

```
library(e1071)
library(kernlab)

variables <- train_data %>% select(F365W:F814W)
dim(variables)
label <- train_data %>% select(galaxy)
dim(label)

variables_test <- test_data %>% select(F365W:F814W)
dim(variables_test)
label_test <- test_data %>% select(galaxy)
dim(label_test)

## [1] 19296     24
## [1] 19296      1
## [1] 8269      24
## [1] 8269      1

model = svm(y = label, x = variables, kernel = "linear", cost = 10,
            type = "C-classification", scale = FALSE)
```

```

summary(model)

##
## Call:
## svm.default(x = variables, y = label, scale = FALSE, type = "C-classification",
##           kernel = "linear", cost = 10)
##
##
## Parameters:
##   SVM-Type: C-classification
##   SVM-Kernel: linear
##   cost: 10
##   gamma: 0.04166667
##
## Number of Support Vectors: 2940
##
## ( 1474 1466 )
##
##
## Number of Classes: 2
##
## Levels:
## 0 1

```

Tenemos alrededor de 3000 vectores soporte, es decir, objetos que intervienen en el cálculo del hiperplano separador (aproximadamente la mitad de cada grupo). Teniendo en cuenta que contamos con casi 20 mil, el número de vectores soportes es bastante óptimo.

Hallamos el índice de aciertos para el SVM.

```

trainprediction = predict(model, train_data[,2:25], decision.values = TRUE)
tsvm <- table(true = train_data$galaxy, trainprediction)
tsvm
cat('\n')
testprediction = predict(model, test_data[,2:25], decision.values = TRUE)
table(true = test_data$galaxy, testprediction)

##      trainprediction
## true      0      1
## 0  6061   392
## 1   494 12349
##
##      testprediction
## true      0      1
## 0 2557   195
## 1   200 5317

```

## 12.1 SVM no lineal

Ahora probaremos con un modelo SVM con kernel RBF que permite crear hiperplanos no lineales.

```

model_RBF = svm(y = train_data$galaxy, x = train_data[,2:25], kernel = "radial", cost = 1, type = "C-cl
summary(model_RBF)

##

```

```

## Call:
## svm.default(x = train_data[, 2:25], y = train_data$galaxy, scale = FALSE,
##              type = "C-classification", kernel = "radial", cost = 1)
##
## Parameters:
##   SVM-Type: C-classification
##   SVM-Kernel: radial
##   cost: 1
##   gamma: 0.04166667
##
## Number of Support Vectors: 2722
##
##  ( 1364 1358 )
##
## Number of Classes: 2
##
## Levels:
##  0 1

```

Vemos que en comparación con el lineal, el número de vectores soporte se ha reducido. Teniendo en cuenta que ya era un número óptimo esto no provoca gran diferencia. Comprobaremos aún así el índice de aciertos.

```

trainpredictionRBF = predict(model_RBF, train_data[,2:25], decision.values = TRUE)
tsvmRBF <- table(true = train_data$galaxy, trainpredictionRBF)
tsvmRBF
cat('\n')
testpredictionRBF = predict(model_RBF, test_data[,2:25], decision.values = TRUE)
table(true = test_data$galaxy, testpredictionRBF)

```

```

##      trainpredictionRBF
## true      0      1
## 0  6199    254
## 1  231 12612
##
##      testpredictionRBF
## true      0      1
## 0 2637    115
## 1 122 5395

```

Parece que el RBF mejora ligeramente sobre el lineal.

Calculamos ahora nuestras medidas de precisión (TP y FP rate) para ambos.

```

total_pos <- tsvm[2]+tsvm[4]
TP_svm <- tsvm[4]/total_pos
total_neg <- tsvm[1]+tsvm[3]
FP_svm <- tsvm[3]/total_neg
cat("SVM Lineal \n\n")
FP_svm
TP_svm

## SVM Lineal
##
## [1] 0.06074694
## [1] 0.9615355

```

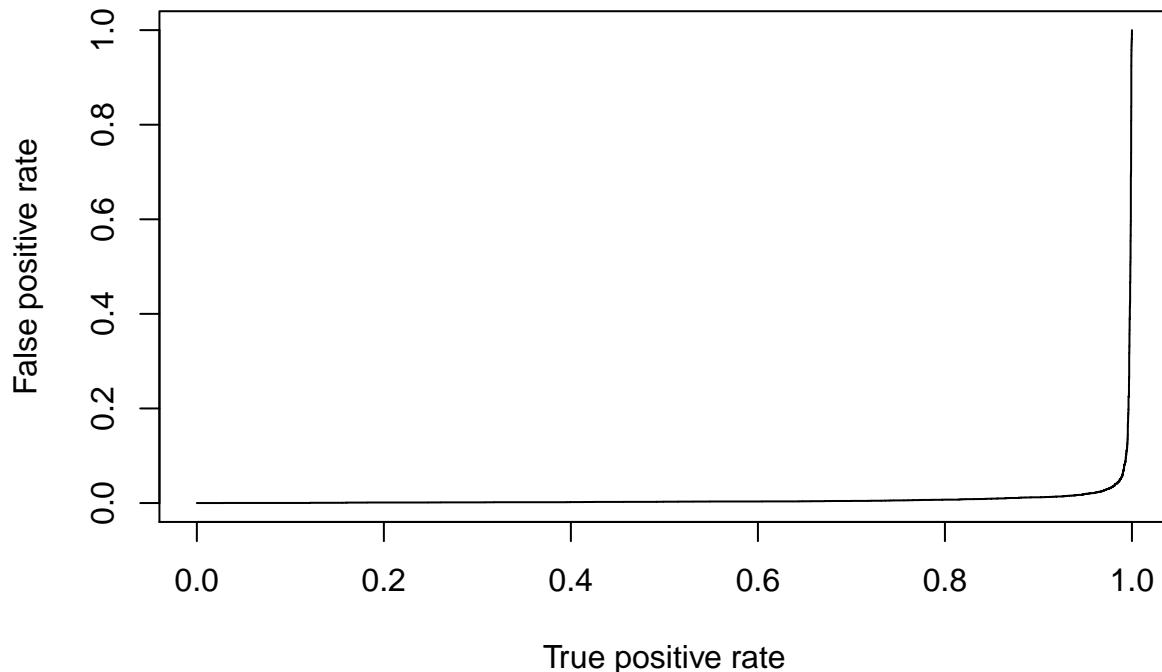
```

total_pos <- tsvmRBF[2]+tsvmRBF[4]
TP_svmRBF <- tsvmRBF[4]/total_pos
total_neg <- tsvmRBF[1]+tsvmRBF[3]
FP_svmRBF <- tsvmRBF[3]/total_neg
cat("SVM No Lineal \n\n")
FP_svmRBF
TP_svmRBF

## SVM No Lineal
##
## [1] 0.03936154
## [1] 0.9820135

predsvm = prediction(attr(trainpredictionRBF, "decision.values"), train_data$galaxy)
predsvm = performance(predsvm, "fpr", "tpr")
plot(predsvm)

```



Aunque hemos visto que el modelo SVM con kernel no lineal da buenos resultados, usaremos la función tune para elegir los mejores parámetros para el modelo.

Pintamos de nuevo la grafica de evaluación de modelos añadiendo los dos modelos SVM:

```

# Como cada realizacion del cluster Kmeans puede variar el grupo, afecta al calculo de False y True pos
par(lab=c(x=8,y=8,len=1))
x <- c(FP_kmean, FP_glm, FP_glm_new)
y <- c(TP_kmean, TP_glm, TP_glm_new)
plot(x, y, pch=3, main="Evaluacion de Modelos", xlab="False Positive Rate",ylab="True Positive Rate", x
par(new="True")

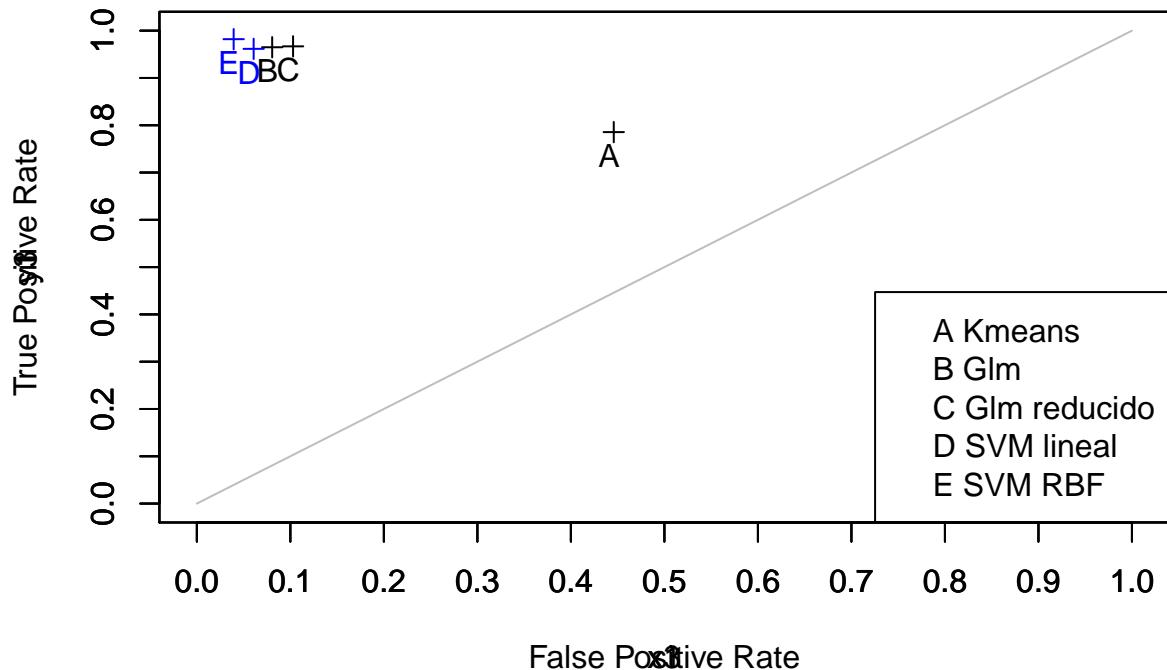
```

```

x1 <- seq(0,1,0.1)
y1 <- x1
plot(x1, y1, type="l", col="grey")
par(new="True")
x3 <- c(FP_svm, FP_svmRBF)
y3 <- c(TP_svm, TP_svmRBF)
X <- c(x, x3)
Y <- c(y, y3)
plot(x3, y3, pch=3, col = "blue", xlim=c(0,1), ylim=c(0,1))
text(X-0.005, Y-0.05, c("A", "B", "C", "D", "E"), col = c("black", "black", "black", "blue", "blue"))
legend("bottomright", legend=c("A Kmeans", "B Glm", "C Glm reducido", "D SVM lineal", "E SVM RBF"))

```

## Evaluacion de Modelos



Lo que nos dice que el SVM mejora las predicciones y, más en concreto, el SVM no lineal.

```

# No hemos conseguido visualizar su resultado debido a que el tiempo de ejecucion no finalizaba. # Esto
# actua mejor en un SVM para nuestro caso
#tuned <- tune.svm(galaxy ~ ., data = variables, gamma = 10^{(-2:2)},
#cost = 10^{(-1:1)})
#summary(tuned)

```

## 13 Estudio con Árboles de decisión

Para explorar otros modelos, como los Árboles de decisión, hemos introducido una columna nueva en el fichero que contiene los objetos clasificados como estrellas (ajustándonos a la clasificación del apartado 5). Este fichero por tanto ya está filtrado con los parámetros de limpieza iniciadas (apartado 2). La columna

añadida es una medida de la temperatura, que se ha añadido como factor, 0 para estrellas frías (por debajo de 3500 K) y estrellas calientes (por encima de 3500 K). Las temperaturas se han obtenido por herramientas de astrofísica para no usar las mismas variables que queremos estudiar en el problema. En este apartado queremos explorar mediante Árboles de decisión la clasificación en el parámetro temperatura.

### 13.1 Nuevo fichero de datos

```
data1 <- read.csv("estrellascontemp.csv")
head(data1)

##          objID      RA      DEC   F365W dF365W   F396W dF396W   F427W dF427W
## 1 81441401246 150.4002  2.0572 21.060  0.009 20.040  0.004 -99.000  0.000
## 2 81462402160 214.3650 52.3448 20.685  0.006 19.982  0.004 19.306  0.003
## 3 81474200192 242.2269 54.7416 21.793  0.207 21.354  0.023 20.392  0.022
## 4 81474300289 242.2427 54.2638 22.237  0.418 21.572  0.089 20.670  0.057
## 5 81474300231 242.2853 54.2653 19.792  0.143 19.368  0.032 18.466  0.022
## 6 81474300201 242.2895 54.2666 21.121  0.107 20.878  0.034 20.552  0.038
##    F458W dF458W   F489W dF489W   F520W dF520W   F551W dF551W   F582W dF582W
## 1    -99       0 -99.000  0.000 -99.000  0.000 -99.000  0.000 -99.000  0.000
## 2    -99       0 -99.000  0.000 -99.000  0.000 -99.000  0.000 -99.000  0.000
## 3    -99       0 19.392  0.009 19.175  0.008 18.706  0.019 18.462  0.007
## 4    -99       0 19.994  0.018 19.793  0.015 19.389  0.016 19.214  0.009
## 5    -99       0 17.644  0.007 17.456  0.006 17.048  0.003 16.856  0.004
## 6    -99       0 20.370  0.020 20.248  0.018 20.204  0.032 20.148  0.017
##    F613W dF613W   F644W dF644W   F675W dF675W   F706W dF706W   F737W
## 1 -99.000  0.000 -99.000  0.000 -99.000  0.000 -99.000  0.000 -99.000
## 2 -99.000  0.000 -99.000  0.000 -99.000  0.000 -99.000  0.000 -99.000
## 3 18.287  0.004 18.095  0.007 18.096  0.007 17.843  0.004 17.760
## 4 19.193  0.004 18.995  0.008 19.018  0.008 18.862  0.006 18.815
## 5 16.806  0.001 16.592  0.003 16.634  0.004 16.465  0.002 16.398
## 6 20.164  0.007 20.077  0.018 20.098  0.018 20.025  0.014 20.075
##    dF737W   F768W dF768W   F799W dF799W   F830W dF830W   F861W dF861W   F892W
## 1 0.000 19.043  0.004 18.958  0.004 19.131  0.003 19.086  0.004 18.629
## 2 0.000 18.673  0.002 19.311  0.002 18.847  0.002 18.610  0.001 18.981
## 3 0.006 -99.000 0.000 17.562  0.004 17.454  0.013 17.408  0.006 17.366
## 4 0.009 18.766  0.027 18.665  0.004 18.639  0.014 18.639  0.013 18.524
## 5 0.003 16.355  0.010 16.252  0.002 16.233  0.006 16.225  0.003 16.141
## 6 0.024 -99.000 0.000 20.008  0.012 20.024  0.041 20.031  0.041 20.024
##    dF892W   F923W dF923W   F954W dF954W      J      dJ      H      dH      KS
## 1 0.004 18.263  0.005 18.203  0.004 19.239  0.008 19.849  0.024 20.279
## 2 0.002 18.771  0.003 18.474  0.006 18.341  0.003 19.316  0.010 20.489
## 3 0.009 17.116  0.015 17.121  0.050 17.164  0.002 16.911  0.004 17.142
## 4 0.011 18.477  0.025 18.840  0.047 18.392  0.008 18.308  0.007 18.698
## 5 0.004 16.048  0.009 16.397  0.015 15.962  0.002 15.903  0.002 16.325
## 6 0.035 19.992  0.075 20.045  0.115 20.070  0.034 20.321  0.038 21.034
##    dKS temp
## 1 0.035    1
## 2 0.030    1
## 3 0.005    1
## 4 0.017    1
## 5 0.003    1
## 6 0.125    1
```

## 13.2 Selección de datos útiles

```
ut <- data1 %>% select(objID,RA, DEC, matches("^F.*W$"),J, H, KS, starts_with("d"),temp)

util_datat <- ut %>% select(objID:F954W,J,H,KS,temp)
head(util_datat)

##          objID        RA       DEC   F365W   F396W   F427W   F458W   F489W   F520W
## 1 81441401246 150.4002  2.0572 21.060 20.040 -99.000    -99 -99.000 -99.000
## 2 81462402160 214.3650 52.3448 20.685 19.982 19.306    -99 -99.000 -99.000
## 3 81474200192 242.2269 54.7416 21.793 21.354 20.392    -99 19.392 19.175
## 4 81474300289 242.2427 54.2638 22.237 21.572 20.670    -99 19.994 19.793
## 5 81474300231 242.2853 54.2653 19.792 19.368 18.466    -99 17.644 17.456
## 6 81474300201 242.2895 54.2666 21.121 20.878 20.552    -99 20.370 20.248
##      F551W   F582W   F613W   F644W   F675W   F706W   F737W   F768W   F799W
## 1 -99.000 -99.000 -99.000 -99.000 -99.000 -99.000 -99.000 19.043 18.958
## 2 -99.000 -99.000 -99.000 -99.000 -99.000 -99.000 -99.000 18.673 19.311
## 3 18.706 18.462 18.287 18.095 18.096 17.843 17.760 -99.000 17.562
## 4 19.389 19.214 19.193 18.995 19.018 18.862 18.815 18.766 18.665
## 5 17.048 16.856 16.806 16.592 16.634 16.465 16.398 16.355 16.252
## 6 20.204 20.148 20.164 20.077 20.098 20.025 20.075 -99.000 20.008
##      F830W   F861W   F892W   F923W   F954W        J        H        KS temp
## 1 19.131 19.086 18.629 18.263 18.203 19.239 19.849 20.279     1
## 2 18.847 18.610 18.981 18.771 18.474 18.341 19.316 20.489     1
## 3 17.454 17.408 17.366 17.116 17.121 17.164 16.911 17.142     1
## 4 18.639 18.639 18.524 18.477 18.840 18.392 18.308 18.698     1
## 5 16.233 16.225 16.141 16.048 16.397 15.962 15.903 16.325     1
## 6 20.024 20.031 20.024 19.992 20.045 20.070 20.321 21.034     1
```

## 13.3 Limpieza

```
#Convertir -99 en NA
util_datat[c(4:27)][(util_datat[,c(4:27)] == -99)] <- NA
#Asignacion de limites
util_datat$F365W[(util_datat$F365W == 99) | (util_datat$F365W > 25.2)] <- 25.2
util_datat$F396W[(util_datat$F396W == 99) | (util_datat$F396W > 25.2)] <- 25.2
util_datat$F427W[(util_datat$F427W == 99) | (util_datat$F427W > 25.2)] <- 25.2
util_datat$F458W[(util_datat$F458W == 99) | (util_datat$F458W > 25.2)] <- 25.2
util_datat$F489W[(util_datat$F489W == 99) | (util_datat$F489W > 25.2)] <- 25.2
util_datat$F520W[(util_datat$F520W == 99) | (util_datat$F520W > 25.0)] <- 25.0
util_datat$F551W[(util_datat$F551W == 99) | (util_datat$F551W > 24.9)] <- 24.9
util_datat$F582W[(util_datat$F582W == 99) | (util_datat$F582W > 24.8)] <- 24.8
util_datat$F613W[(util_datat$F613W == 99) | (util_datat$F613W > 24.8)] <- 24.8
util_datat$F644W[(util_datat$F644W == 99) | (util_datat$F644W > 24.7)] <- 24.7
util_datat$F675W[(util_datat$F675W == 99) | (util_datat$F675W > 24.7)] <- 24.7
util_datat$F706W[(util_datat$F706W == 99) | (util_datat$F706W > 24.7)] <- 24.7
util_datat$F737W[(util_datat$F737W == 99) | (util_datat$F737W > 24.6)] <- 24.6
util_datat$F768W[(util_datat$F768W == 99) | (util_datat$F768W > 24.5)] <- 24.5
util_datat$F799W[(util_datat$F799W == 99) | (util_datat$F799W > 24.5)] <- 24.5
util_datat$F830W[(util_datat$F830W == 99) | (util_datat$F830W > 24.3)] <- 24.3
util_datat$F861W[(util_datat$F861W == 99) | (util_datat$F861W > 24.3)] <- 24.3
util_datat$F892W[(util_datat$F892W == 99) | (util_datat$F892W > 24.1)] <- 24.1
```

```

util_datat$F923W[(util_datat$F923W == 99) | (util_datat$F923W > 23.9)] <- 23.9
util_datat$F954W[(util_datat$F954W == 99) | (util_datat$F954W > 23.4)] <- 23.4
util_datat$J[(util_datat$J == 99) | (util_datat$J > 24.0)] <- 24.0
util_datat$H[(util_datat$H == 99) | (util_datat$H > 23.6)] <- 23.6
util_datat$KS[(util_datat$KS == 99) | (util_datat$KS > 23.4)] <- 23.4

head(util_datat)

##          objID      RA     DEC   F365W   F396W   F427W   F458W   F489W   F520W
## 1 81441401246 150.4002 2.0572 21.060 20.040      NA      NA      NA      NA
## 2 81462402160 214.3650 52.3448 20.685 19.982 19.306      NA      NA      NA
## 3 81474200192 242.2269 54.7416 21.793 21.354 20.392      NA 19.392 19.175
## 4 81474300289 242.2427 54.2638 22.237 21.572 20.670      NA 19.994 19.793
## 5 81474300231 242.2853 54.2653 19.792 19.368 18.466      NA 17.644 17.456
## 6 81474300201 242.2895 54.2666 21.121 20.878 20.552      NA 20.370 20.248
##          F551W   F582W   F613W   F644W   F675W   F706W   F737W   F768W   F799W   F830W
## 1        NA      NA      NA      NA      NA      NA 19.043 18.958 19.131
## 2        NA      NA      NA      NA      NA      NA 18.673 19.311 18.847
## 3 18.706 18.462 18.287 18.095 18.096 17.843 17.760      NA 17.562 17.454
## 4 19.389 19.214 19.193 18.995 19.018 18.862 18.815 18.766 18.665 18.639
## 5 17.048 16.856 16.806 16.592 16.634 16.465 16.398 16.355 16.252 16.233
## 6 20.204 20.148 20.164 20.077 20.098 20.025 20.075      NA 20.008 20.024
##          F861W   F892W   F923W   F954W      J      H      KS temp
## 1 19.086 18.629 18.263 18.203 19.239 19.849 20.279      1
## 2 18.610 18.981 18.771 18.474 18.341 19.316 20.489      1
## 3 17.408 17.366 17.116 17.121 17.164 16.911 17.142      1
## 4 18.639 18.524 18.477 18.840 18.392 18.308 18.698      1
## 5 16.225 16.141 16.048 16.397 15.962 15.903 16.325      1
## 6 20.031 20.024 19.992 20.045 20.070 20.321 21.034      1

```

### 13.4 Ficheros de entrenamiento y testeo

```

n_datat=dim(util_datat)[1]

n_traint=round(0.7*n_datat)
n_testt=n_datat-n_traint

indicest=1:n_datat
indices_traint= sample(indicest,n_traint)
indices_testt=indicest[-indices_traint]

train_datat=util_datat[indices_traint,]
test_datat=util_datat[indices_testt,]
dim(train_datat)
dim(test_datat)
class(train_datat$temp)
head(train_datat)

## [1] 7561    27
## [1] 3241    27
## [1] "integer"
##          objID      RA     DEC   F365W   F396W   F427W   F458W   F489W
## 7685 81474305315 242.5906 54.1114 25.200 24.267 23.785 23.194 23.179

```

```

## 8886 81422303439 36.8500 0.6782 25.200 25.200 25.200 24.327 24.080
## 5817 81451308213 188.4195 61.7614 23.547 23.381 22.730 22.130 21.947
## 4182 81431404831 139.5556 45.9116 23.243 22.954 21.894 21.079 20.932
## 98 81481104085 356.9562 15.9671 17.389 17.027 16.803 16.525 16.443
## 6283 81462404822 214.4324 52.2683 24.215 23.723 22.938 22.389 22.179
## F520W F551W F582W F613W F644W F675W F706W F737W F768W F799W
## 7685 22.764 22.326 22.102 21.980 21.724 21.775 21.497 21.264 21.157 21.076
## 8886 23.849 23.433 23.197 22.935 22.747 22.642 22.480 22.455 22.203 22.071
## 5817 21.871 21.663 21.566 21.332 21.277 21.175 21.088 21.047 21.006 20.980
## 4182 20.662 20.297 20.063 19.980 19.661 19.654 19.480 19.287 19.132 19.035
## 98 16.369 16.392 16.346 16.348 16.311 16.293 16.296 16.175 16.203 16.213
## 6283 22.112 21.790 21.578 21.461 21.364 21.254 21.177 21.028 21.058 21.010
## F830W F861W F892W F923W F954W J H KS temp
## 7685 20.989 20.980 20.815 20.706 20.862 20.579 20.455 20.973 1
## 8886 22.042 21.995 21.613 22.012 21.397 21.513 21.568 21.912 1
## 5817 20.935 20.877 20.863 20.781 20.907 20.696 20.727 21.055 1
## 4182 18.965 18.962 18.895 18.849 18.961 18.630 18.473 18.799 1
## 98 16.180 16.245 16.243 16.173 16.408 16.332 16.582 17.102 1
## 6283 20.965 20.919 20.937 20.824 20.767 20.672 20.620 20.874 1

```

### 13.5 Regresión logística para estudio de dependencia de variables

```

glm_temp=glm(temp~F365W+F396W+F427W+F458W+F489W+F520W+F551W+F582W+
             F613W+F644W+F675W+F706W+F737W+F768W+F799W+
             F830W+F861W+F892W+F923W+J+H+KS,
             data = train_datat)
summary(glm_temp)

##
## Call:
## glm(formula = temp ~ F365W + F396W + F427W + F458W + F489W +
##      F520W + F551W + F582W + F613W + F644W + F675W + F706W + F737W +
##      F768W + F799W + F830W + F861W + F892W + F923W + J + H + KS,
##      data = train_datat)
##
## Deviance Residuals:
##       Min        1Q     Median        3Q       Max
## -1.31200 -0.05083 -0.00467  0.07119  1.84860
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)    
## (Intercept) 0.867761  0.033691 25.756 < 2e-16 ***
## F365W      -0.041986  0.006391 -6.569 5.40e-11 ***
## F396W       0.052481  0.009656  5.435 5.65e-08 ***
## F427W      -0.047999  0.012126 -3.958 7.62e-05 ***
## F458W      -0.035639  0.017140 -2.079  0.03762 *  
## F489W       0.132270  0.022519  5.874 4.45e-09 ***
## F520W       0.321268  0.027875 11.526 < 2e-16 ***
## F551W       0.020660  0.026519  0.779  0.43596    
## F582W      -0.362713  0.026562 -13.655 < 2e-16 ***
## F613W      -0.274871  0.024866 -11.054 < 2e-16 ***
## F644W       0.061549  0.037655  1.635  0.10218    
## F675W      -0.158806  0.030860 -5.146 2.73e-07 ***

```

```

## F706W      -0.109910  0.040990 -2.681  0.00735  **
## F737W       0.170244  0.028261  6.024  1.78e-09 ***
## F768W      -0.397448  0.045267 -8.780  < 2e-16 ***
## F799W       0.497328  0.043431 11.451  < 2e-16 ***
## F830W      -0.168966  0.037989 -4.448  8.81e-06 ***
## F861W        0.192515  0.044271  4.349  1.39e-05 ***
## F892W       0.081667  0.034692  2.354  0.01860  *
## F923W       0.032254  0.025227  1.279  0.20110
## J           0.096814  0.030138  3.212  0.00132  **
## H           -0.044990  0.019216 -2.341  0.01925  *
## KS          -0.006948  0.010327 -0.673  0.50113
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for gaussian family taken to be 0.04093482)
##
## Null deviance: 965.86  on 7403  degrees of freedom
## Residual deviance: 302.14  on 7381  degrees of freedom
##   (157 observations deleted due to missingness)
## AIC: -2624.9
##
## Number of Fisher Scoring iterations: 2

```

## 13.6 Árbol de decisión

Eliminando las variables que salen sin dependencia en el apartado anterior (F644W, F923W y KS), construimos un Árbol de decisión. Establecemos inicialmente un factor de coste de 0.001, el método de clasificación y no ponemos límite de profundidad.

```

library(rpart)
set.seed(123)

dfrpt <- rpart(temp~F365W+F396W+F427W+F458W+F489W+F520W+F582W+
                 F613W+F675W+F737W+F768W+F799W+
                 F830W+F861W+J, data=train_datat,cp=0.001,parms=list(split="information"))
dfrpt

## n= 7561
##
## node), split, n, deviance, yval
##      * denotes terminal node
##
##    1) root 7561 983.4199000 0.84631660
##    2) J< 18.8655 3270 708.8713000 0.68226300
##      4) F427W>=21.985 857 128.8705000 0.18436410
##      8) F427W>=22.504 551 26.5771300 0.05081670
##      16) F613W>=20.5255 452 11.6814200 0.02654867 *
##      17) F613W< 20.5255 99 13.4141400 0.16161620
##      34) F799W< 18.978 85 2.8941180 0.03529412 *
##      35) F799W>=18.978 14 0.9285714 0.92857140 *
##      9) F427W< 22.504 306 74.7712400 0.42483660
##      18) J< 18.127 204 28.9951000 0.17156860
##      36) J< 17.7735 130 3.8769230 0.03076923 *
##      37) J>=17.7735 74 18.0135100 0.41891890

```

```

##          74) F582W>=20.265 37    2.7567570 0.08108108 *
##          75) F582W< 20.265 37    6.8108110 0.75675680
##          150) F520W>=20.6635 13    3.2307690 0.46153850 *
##          151) F520W< 20.6635 24    1.8333330 0.91666670 *
##          19) J>=18.127 102    6.5196080 0.93137250
##          38) F613W>=20.4925 9     2.0000000 0.33333330 *
##          39) F613W< 20.4925 93    0.9892473 0.98924730 *
##          5) F427W< 21.985 2413   292.0928000 0.85909660
##          10) F427W>=20.7115 800   183.7550000 0.64250000
##          20) J< 17.257 288    50.8750000 0.22916670
##          40) F427W>=21.11 171    12.0117000 0.07602339
##          80) J< 17.174 143    4.8251750 0.03496503 *
##          81) J>=17.174 28    5.7142860 0.28571430
##          162) F520W>=19.873 21   1.8095240 0.09523810 *
##          163) F520W< 19.873 7    0.8571429 0.85714290 *
##          41) F427W< 21.11 117    28.9914500 0.45299150
##          82) J< 16.7955 68    7.8088240 0.13235290
##          164) F830W< 17.029 39   0.9743590 0.02564103 *
##          165) F830W>=17.029 29   5.7931030 0.27586210
##          330) F489W>=19.7955 19   1.7894740 0.10526320 *
##          331) F489W< 19.7955 10   2.4000000 0.60000000 *
##          83) J>=16.7955 49    4.4897960 0.89795920 *
##          21) J>=17.257 512    56.0000000 0.87500000
##          42) J< 17.8055 203    42.2660100 0.70443350
##          84) F582W>=19.7115 60   10.7333300 0.23333330
##          168) F365W< 22.96 19    0.0000000 0.00000000 *
##          169) F365W>=22.96 41    9.2195120 0.34146340
##          338) F427W>=21.8245 22   2.5909090 0.13636360 *
##          339) F427W< 21.8245 19   4.6315790 0.57894740 *
##          85) F582W< 19.7115 143   12.6293700 0.90209790
##          170) F427W>=21.5555 24   5.8333330 0.58333330
##          340) F861W< 17.9905 16   3.7500000 0.37500000 *
##          341) F861W>=17.9905 8    0.0000000 1.00000000 *
##          171) F427W< 21.5555 119   3.8655460 0.96638660 *
##          43) J>=17.8055 309    3.9482200 0.98705500 *
##          11) F427W< 20.7115 1613   52.1921900 0.96652200
##          22) J< 15.6585 83    18.5542200 0.66265060
##          44) F489W>=18.45 31    3.4838710 0.12903230 *
##          45) F489W< 18.45 52    0.9807692 0.98076920 *
##          23) J>=15.6585 1530   25.5581700 0.98300650
##          46) F427W>=20.253 240   21.6000000 0.90000000
##          92) J< 16.0425 12    1.6666670 0.16666670 *
##          93) J>=16.0425 228   13.1403500 0.93859650
##          186) J< 16.6175 51    9.1764710 0.76470590
##          372) F489W>=19.38 30   7.2000000 0.60000000 *
##          373) F489W< 19.38 21   0.0000000 1.00000000 *
##          187) J>=16.6175 177   1.9774010 0.98870060 *
##          47) F427W< 20.253 1290   1.9968990 0.99844960 *
##          3) J>=18.8655 4291   119.4742000 0.97133540
##          6) F613W>=23.5735 55   7.5272730 0.16363640
##          12) F799W< 21.2515 48   2.8125000 0.06250000
##          24) F427W>=24.742 41   0.0000000 0.00000000 *
##          25) F427W< 24.742 7    1.7142860 0.42857140 *
##          13) F799W>=21.2515 7   0.8571429 0.85714290 *

```

```

##    7) F613W< 23.5735 4236  75.6003300 0.98182250
##    14) F613W>=23.16 122  18.6639300 0.81147540
##      28) F737W< 21.925 22  0.0000000 0.00000000 *
##      29) F737W>=21.925 100  0.9900000 0.99000000 *
##    15) F613W< 23.16 4114  53.2912000 0.98687410
##      30) F458W>=23.4755 692  35.9132900 0.94508670
##        60) F799W< 20.834 17  0.9411765 0.05882353 *
##        61) F799W>=20.834 675  21.2829600 0.96740740 *
##    31) F458W< 23.4755 3422  15.9251900 0.99532440 *

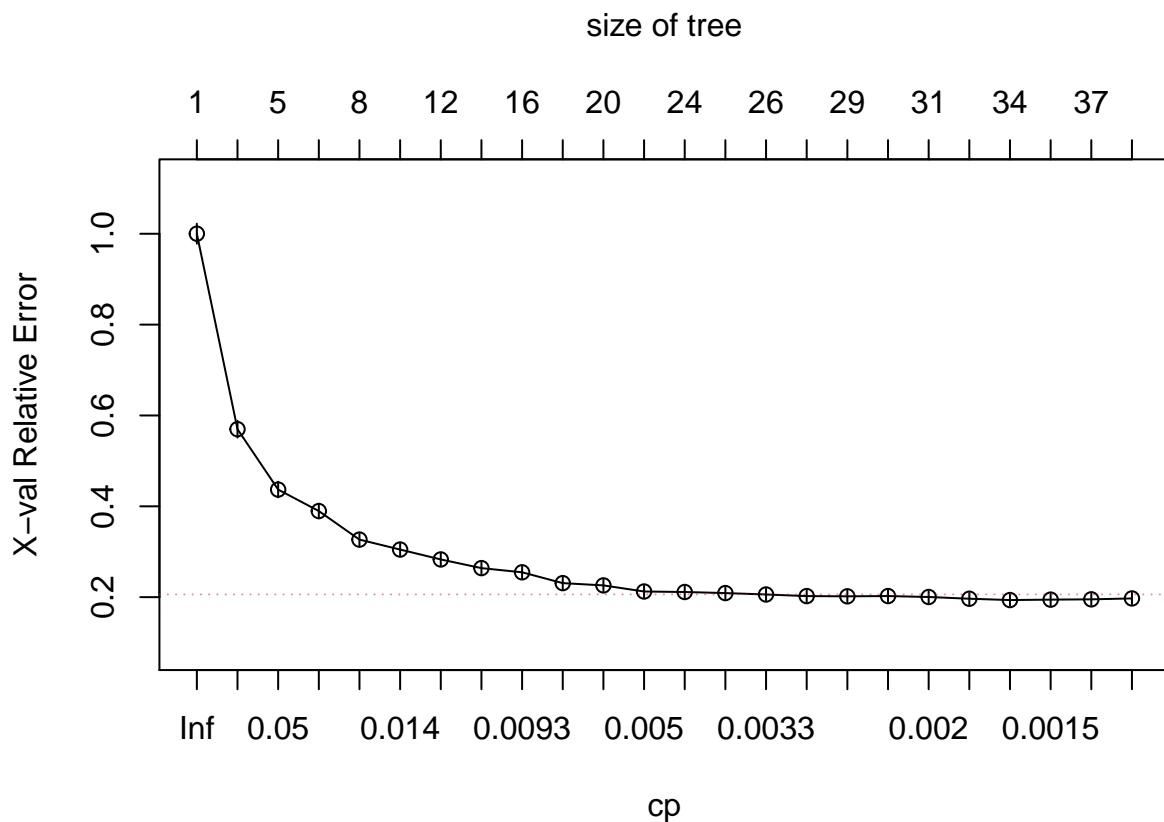
```

Pintamos las gráficas del Árbol y del factor de coste.

```

library(rpart.plot)
library(RColorBrewer)
labels(dfrpt, pretty=T)
plotcp(dfrpt)

```

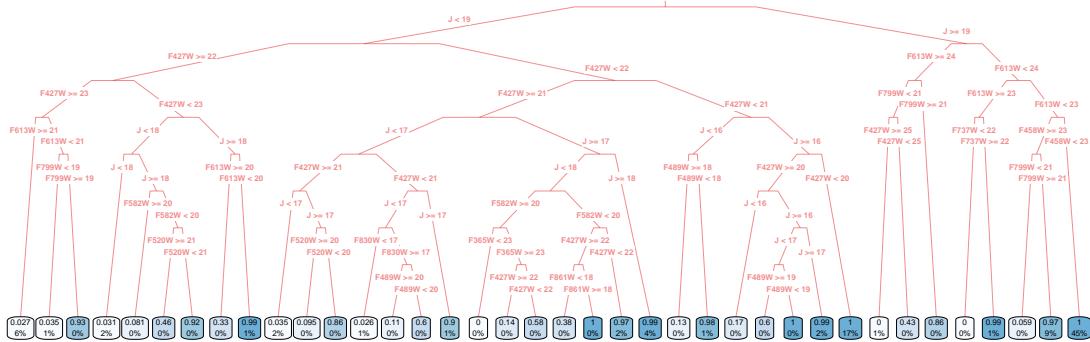


```

printcp(dfrpt)
rpart.plot(dfrpt, type=3, branch=0.3, clip.right.labs=FALSE, main="Arbol de clasificación")

```

## Arbol de clasificación



```

## [1] "root"           "J< 18.87"       "F427W>=21.98" "F427W>=22.5"
## [5] "F613W>=20.53" "F613W< 20.53" "F799W< 18.98" "F799W>=18.98"
## [9] "F427W< 22.5"  "J< 18.13"       "J< 17.77"      "J>=17.77"
## [13] "F582W>=20.27" "F582W< 20.27" "F520W>=20.66" "F520W< 20.66"
## [17] "J>=18.13"     "F613W>=20.49" "F613W< 20.49" "F427W< 21.98"
## [21] "F427W>=20.71" "J< 17.26"       "F427W>=21.11" "J< 17.17"
## [25] "J>=17.17"     "F520W>=19.87" "F520W< 19.87" "F427W< 21.11"
## [29] "J< 16.8"       "F830W< 17.03" "F830W>=17.03" "F489W>=19.8"
## [33] "F489W< 19.8"  "J>=16.8"       "J>=17.26"      "J< 17.81"
## [37] "F582W>=19.71" "F365W< 22.96" "F365W>=22.96" "F427W>=21.82"
## [41] "F427W< 21.82" "F582W< 19.71" "F427W>=21.56" "F861W< 17.99"
## [45] "F861W>=17.99" "F427W< 21.56" "J>=17.81"      "F427W< 20.71"
## [49] "J< 15.66"     "F489W>=18.45" "F489W< 18.45" "J>=15.66"
## [53] "F427W>=20.25" "J< 16.04"       "J>=16.04"      "J< 16.62"
## [57] "F489W>=19.38" "F489W< 19.38" "J>=16.62"      "F427W< 20.25"
## [61] "J>=18.87"     "F613W>=23.57" "F799W< 21.25" "F427W>=24.74"
## [65] "F427W< 24.74" "F799W>=21.25" "F613W< 23.57" "F613W>=23.16"
## [69] "F737W< 21.93" "F737W>=21.93" "F613W< 23.16" "F458W>=23.48"
## [73] "F799W< 20.83" "F799W>=20.83" "F458W< 23.48" "#"
## Regression tree:
## rpart(formula = temp ~ F365W + F396W + F427W + F458W + F489W +
##        F520W + F582W + F613W + F675W + F737W + F768W + F799W + F830W +
##        F861W + J, data = train_datat, parms = list(split = "information"),
##        cp = 0.001)
##
```

```

## Variables actually used in tree construction:
## [1] F365W F427W F458W F489W F520W F582W F613W F737W F799W F830W F861W
## [12] J
##
## Root node error: 983.42/7561 = 0.13006
##
## n= 7561
##
##          CP nsplit rel.error xerror      xstd
## 1  0.2252254      0  1.00000 1.00014 0.022090
## 2  0.0676342      2  0.54955 0.56981 0.018512
## 3  0.0369594      4  0.41428 0.43686 0.017300
## 4  0.0339523      5  0.37732 0.38958 0.016278
## 5  0.0145864      7  0.30942 0.32674 0.015418
## 6  0.0135063      9  0.28024 0.30481 0.015151
## 7  0.0112716     11  0.25323 0.28301 0.014735
## 8  0.0108393     13  0.23069 0.26393 0.014400
## 9  0.0079064     15  0.20901 0.25474 0.014122
## 10 0.0076986     17  0.19320 0.23090 0.013532
## 11 0.0056299     19  0.17780 0.22593 0.013398
## 12 0.0044509     21  0.16654 0.21249 0.013012
## 13 0.0039227     23  0.15764 0.21133 0.013012
## 14 0.0035899     24  0.15372 0.20901 0.012975
## 15 0.0029799     25  0.15013 0.20592 0.012930
## 16 0.0022980     26  0.14715 0.20236 0.012762
## 17 0.0021185     28  0.14255 0.20182 0.012828
## 18 0.0020200     29  0.14043 0.20243 0.012845
## 19 0.0020098     30  0.13841 0.20041 0.012765
## 20 0.0017850     31  0.13640 0.19654 0.012623
## 21 0.0017762     33  0.13283 0.19363 0.012463
## 22 0.0013448     34  0.13106 0.19454 0.012504
## 23 0.0011167     36  0.12837 0.19520 0.012493
## 24 0.0010000     37  0.12725 0.19723 0.012608

```

De la tabla del factor de coste, con nuestro cp inicial, se ve que el valor que minimiza el error es 0.00177, muy similar al usado inicialmente aunque mayor. Aún así, usamos este valor extrayéndolo directamente de la tabla para *podar* el Árbol.

```

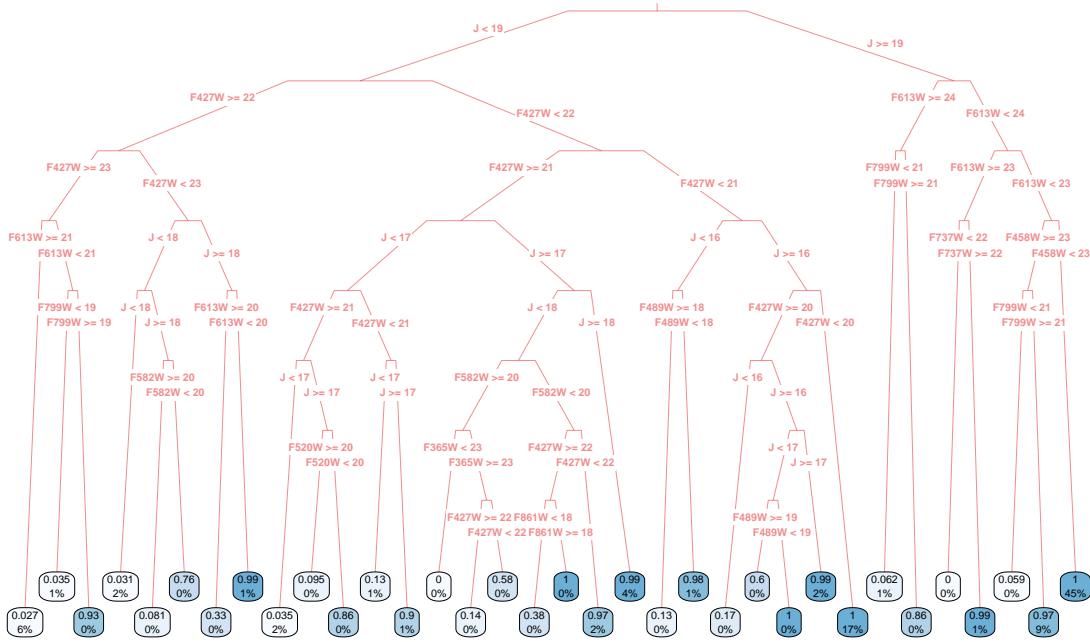
#Poda
poda_dfrpt<- prune(dfrpt, cp=dfrpt$cptable[which.min(dfrpt$cptable[, "xerror"]),"CP"])

#pintamos arbol podado

rpart.plot(poda_dfrpt,type=3,branch=0.3,clip.right.labs=FALSE,
           main="Arbol de clasificacion podado")

```

## Arbol de clasificación podado

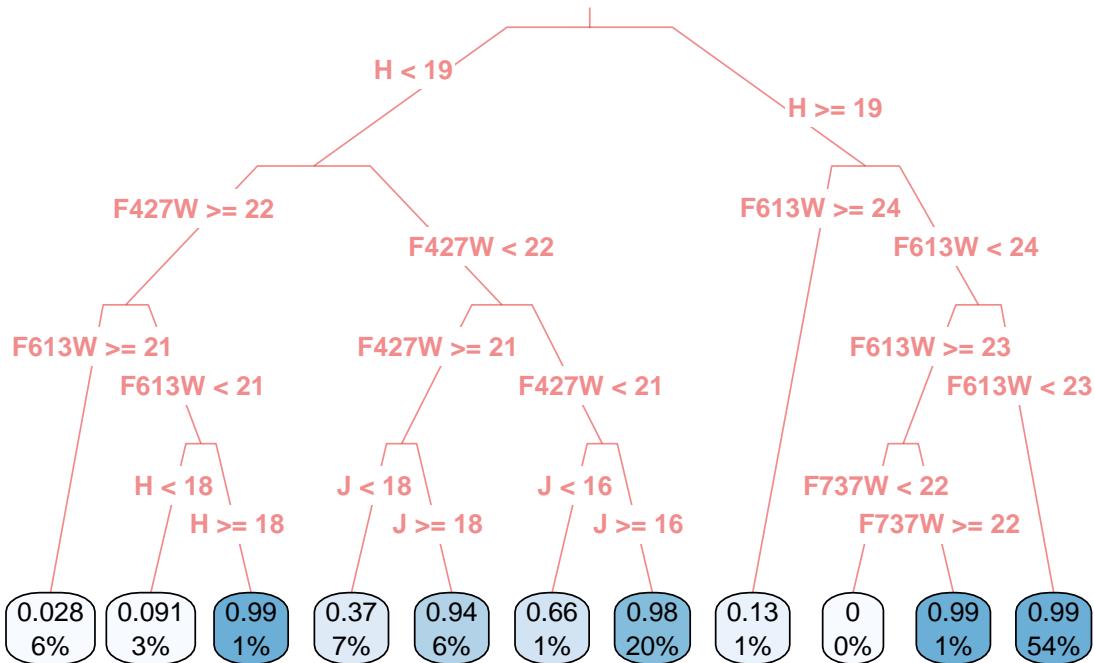


Solo para mejor visualización, ya que el Árbol que sale es demasiado grande, realizamos el mismo ejercicio con una profundidad máxima de 4 y dibujamos.

```
dfrpt2 <- rpart(temp~F365W+F396W+F427W+F458W+F489W+F520W+F551W+F582W+
F613W+F675W+F706W+F737W+F768W+F799W+
F830W+F861W+F892W+J+H, data=train_datat, cp=0.0017762,
parms=list(split="information"),control=list(maxdepth=4))

rpart.plot(dfrpt2,type=3,branch=0.3,clip.right.labs=FALSE,main=
"Arbol de clasificación cortado a profundidad 4")
```

## Arbol de clasificacion cortado a profundidad 4

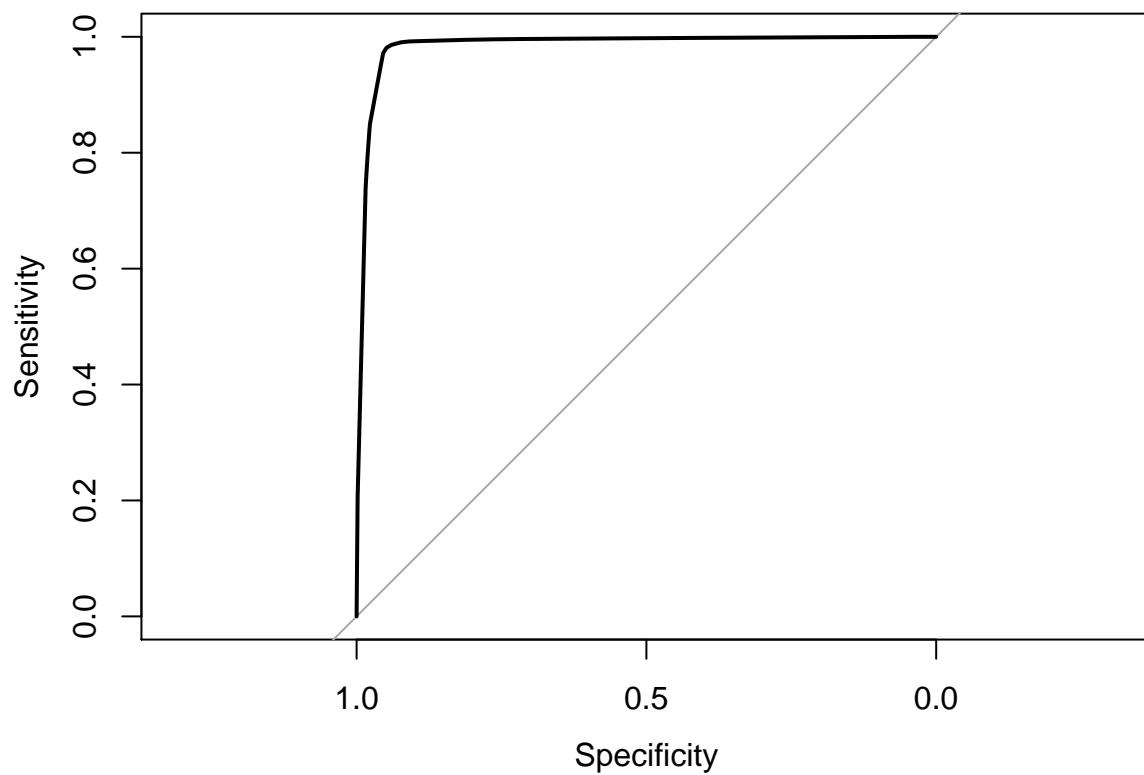


Predecimos sobre los datos train y sobre los datos de test con el modelo completo podado. Añadimos tabla de confusión y curva Roc.

```

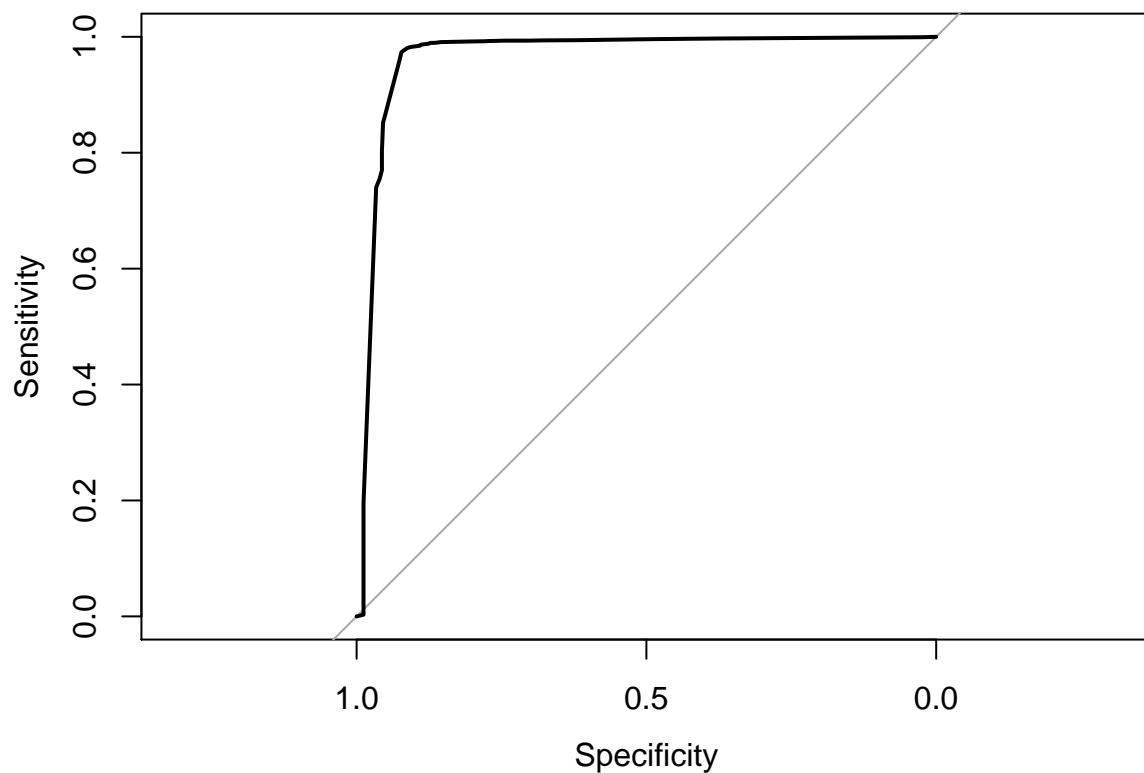
#Para train
n=dim(train_datat[1])
y.pred=predict(poda_dfrpt,train_datat[,-c(25)])
#y.pred

table(train_datat$temp, y.pred > 0.8)
tr_train <- roc(temp ~ y.pred, data = train_datat)
plot(tr_train)
  
```



```
#Para test
n2=dim(test_datat[1])
y2.pred=predict(poda_dfrpt,test_datat[,-c(25)])
#y2.pred

table(test_datat$temp, y2.pred > 0.8)
tr_test <- roc(temp ~ y2.pred, data = test_datat)
plot(tr_test)
```



```
##          FALSE TRUE
## 0    1101   61
## 1     117 6282
##
##          FALSE TRUE
## 0     457   47
## 1      48 2689
```

Precisión del modelo

```
# Area bajo al curva de ROC

#train
auc_DT_ROCt = auc(train_datat$temp,y.pred)
auc_DT_ROCt

#test
auc_DT_ROC = auc(test_datat$temp,y2.pred)
auc_DT_ROC

## Area under the curve: 0.9848
## Area under the curve: 0.9666
```

Comprobamos usando de nuevo un índice astrofísico que separa temperatura como se distribuyen los datos de la predicción:

```

tt=data.frame(test_datat,tempred=y2.pred)

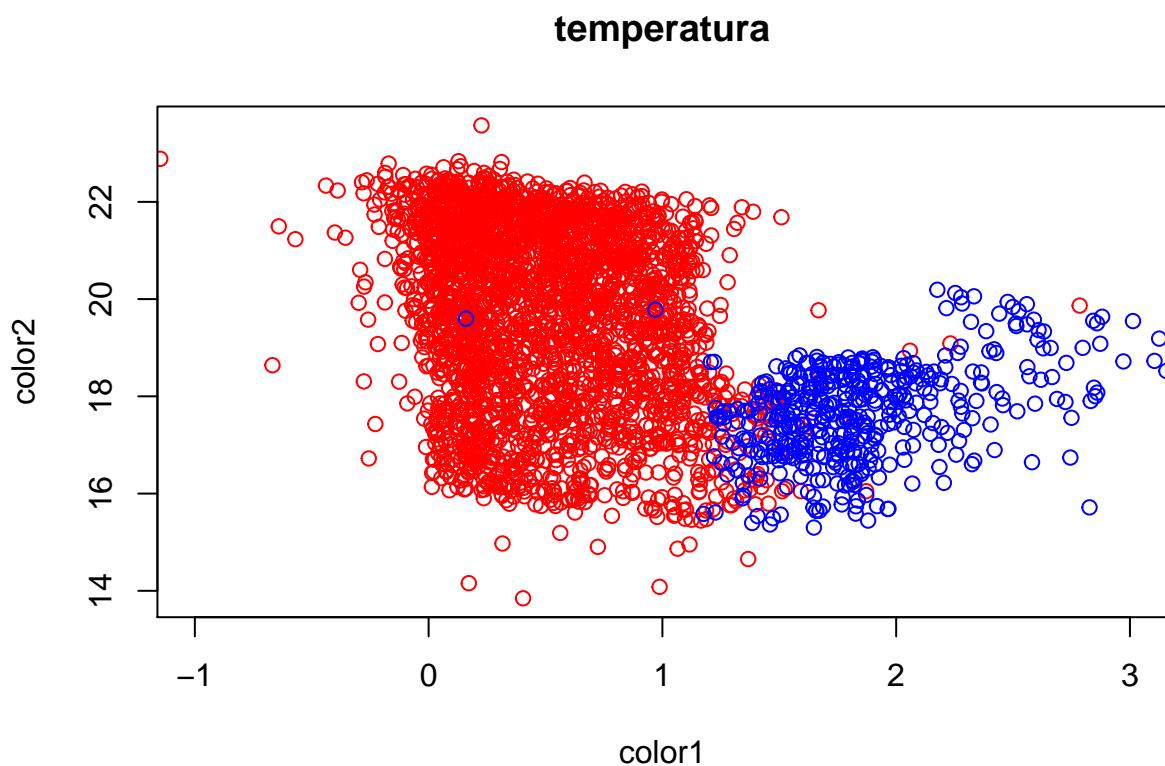
#Separamos en datos frios y calientes segun la prediccion para dibujar

ttc= tt  %>% filter(tempred>0.5) #calientes en rojo
ttf= tt  %>% filter(tempred<0.5) #frias en azul

xttc=ttc$F644W-ttc$F923W
xttf=ttf$F644W-ttf$F923W

plot(xttc,ttc$J,col='red',main="temperatura",xlab="color1",ylab="color2",xlim=c(-1,3))
points(xttf,ttf$J,col='blue')

```



## 14 Resumen y conclusiones

Los objetivos eran:

- 1). Separar objetos de tipo galáctico y estelar.
- 2). Discriminar objetos estelares por temperatura.

En ambos casos se han conseguido resultados similares a procedimientos clásicos de astrofísica, usando **regresión logística**, **máquinas de vector soporte** y **k-means** en el problema 1), y **árboles de decisión** en el 2).

Todos los métodos usados han dado buen resultado aunque, como mencionamos en el trimestre pasado, el método de clusterización K means elegido tal vez no era el más adecuado.

Los datos se componen de datos fotométricos a lo largo de un rango extenso en longitud de onda. Normalmente para hacer la clasificación entre objetos de distinta naturaleza (estrellas y galaxias por ejemplo), se usan combinaciones de colores (restas de datos fotométricos a distinta longitud de onda), que se han ido seleccionando y perfeccionando con el tiempo y que se revisan para cada catálogo.

Los métodos usados en esta práctica son igual de eficientes y llegan a similar conclusión sin conocimiento previo. Que los modelos de regresión simplificados sean equivalentes es también normal, puesto que no todos los índices sirven para discriminar todos los tipos de objetos. Y en este caso no se han realizado combinaciones de datos.

En el caso de añadir la temperatura, igualmente se usan índices de color para separar por comparación de color, si unos objetos son más fríos o más calientes que otros y siguen el patrón de los de antes, es decir, búsqueda de la mejor combinación a lo largo de los años. Un caso que se podría haber implementado con redes neuronales pero que no teníamos tiempo de hacer en R, habría sido determinar la temperatura numérica. Normalmente con tantos datos de fotometría, se puede comparar con modelos y obtener la temperatura. Creemos que con una red neuronal podríamos haber entrenado los datos para determinar la temperatura de los objetos.

Los métodos usados en esta práctica empiezan a popularizarse en astrofísica para facilitar y agilizar trabajos de clasificación de este tipo.