In [1]:
```
data <- read.csv2("SpotifyProjet.csv")
```

In [2]:
```
str(data)
```

```
'data.frame':   116 obs. of  16 variables:
 $ ï..Artist       : Factor w/ 91 levels "070 shake","50 cent",..: 17 3 28 84 87 77 54 19 52 60 ...
 $ Title           : Factor w/ 116 levels "\"2:30\"","\"911\"",..: 50 37 71 47 98 70 9 46 105 67 ...
 $ Genre           : Factor w/ 19 levels "Acoustic","Afro",..: 16 14 17 16 17 1 11 3 16 18 ...
 $ Danceability    : num  0.502 0.749 0.664 0.805 0.645 0.365 0.778 0.782 0.509 0.743 ...
 $ Energy          : num  0.64 0.491 0.609 0.498 0.534 0.273 0.695 0.559 0.544 0.622 ...
 $ Key             : int  6 10 1 7 6 4 4 6 5 4 ...
 $ Loudness        : num  -9.7 -9.65 -6.51 -7.93 -10.8 ...
 $ Mode            : int  0 1 1 0 0 0 0 0 0 0 ...
 $ Speechiness     : num  0.0286 0.0403 0.0707 0.0737 0.0479 0.038 0.0913 0.0767 0.0307 0.136 ...
 $ Acousticness    : num  0.0313 0.02 0.304 0.0203 0.157 0.94 0.175 0.125 0.63 0.904 ...
 $ Instrumentalness: num  2.16e-06 8.99e-03 0.00 2.37e-05 0.00 4.31e-01 0.00 0.00 1.48e-01 1.66e-05 ...
 $ Liveness        : num  0.154 0.159 0.0926 0.085 0.0863 0.109 0.15 0.385 0.1 0.143 ...
 $ Valence         : num  0.449 0.536 0.194 0.636 0.463 0.238 0.472 0.685 0.206 0.317 ...
 $ Duration        : num  211 293 211 214 226 ...
 $ Tempo           : num  132 120 130 121 105 ...
 $ Liked           : Factor w/ 2 levels "No","Yes": 2 2 2 2 2 2 2 2 2 2 ...
```

On enlève les 3 premières colonnes de notre dataframe pour simplifier l'étude

In [47]:

```
#don1 représente l'ensemble des données sans les deux premières colonnes Artist et titre
don1=data[,-(1:3)]
#don représente nos données en enlevant la variable Liked que l'on va transformer en nombre binaire et remettre dans don
don=don1[,-13]
don
```

| Danceability | Energy | Key | Loudness | Mode | Speechiness | Acousticness | Instrumentalness | Liveness | Valence | Duration | Tempo |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.502 | 0.6400 | 6 | -9.702 | 0 | 0.0286 | 0.031300 | 2.16e-06 | 0.1540 | 0.449 | 210.730 | 131.551 |
| 0.749 | 0.4910 | 10 | -9.655 | 1 | 0.0403 | 0.020000 | 8.99e-03 | 0.1590 | 0.536 | 292.613 | 119.528 |
| 0.664 | 0.6090 | 1 | -6.509 | 1 | 0.0707 | 0.304000 | 0.00e+00 | 0.0926 | 0.194 | 210.560 | 130.041 |
| 0.805 | 0.4980 | 7 | -7.927 | 0 | 0.0737 | 0.020300 | 2.37e-05 | 0.0850 | 0.636 | 214.215 | 121.006 |
| 0.645 | 0.5340 | 6 | -10.800 | 0 | 0.0479 | 0.157000 | 0.00e+00 | 0.0863 | 0.463 | 226.467 | 105.020 |
| 0.365 | 0.2730 | 4 | -16.526 | 0 | 0.0380 | 0.940000 | 4.31e-01 | 0.1090 | 0.238 | 248.695 | 132.285 |
| 0.778 | 0.6950 | 4 | -6.865 | 0 | 0.0913 | 0.175000 | 0.00e+00 | 0.1500 | 0.472 | 132.780 | 149.996 |
| 0.782 | 0.5590 | 6 | -7.106 | 0 | 0.0767 | 0.125000 | 0.00e+00 | 0.3850 | 0.685 | 188.344 | 129.992 |
| 0.509 | 0.5440 | 5 | -6.798 | 0 | 0.0307 | 0.630000 | 1.48e-01 | 0.1000 | 0.206 | 241.747 | 71.502 |
| 0.743 | 0.6220 | 4 | -5.332 | 0 | 0.1360 | 0.904000 | 1.66e-05 | 0.1430 | 0.317 | 183.133 | 140.054 |
| 0.501 | 0.0958 | 0 | -15.605 | 1 | 0.0433 | 0.770000 | 0.00e+00 | 0.2980 | 0.204 | 248.808 | 108.741 |
| 0.653 | 0.7020 | 4 | -10.106 | 0 | 0.1170 | 0.023200 | 3.36e-02 | 0.0864 | 0.258 | 211.559 | 169.950 |
| 0.762 | 0.7290 | 6 | -8.076 | 0 | 0.0679 | 0.481000 | 1.37e-04 | 0.1080 | 0.482 | 225.040 | 119.956 |
| 0.657 | 0.6430 | 8 | -6.661 | 0 | 0.0433 | 0.482000 | 2.08e-06 | 0.1890 | 0.394 | 251.670 | 82.516 |
| 0.845 | 0.7170 | 5 | -6.771 | 0 | 0.2110 | 0.666000 | 0.00e+00 | 0.1100 | 0.686 | 160.893 | 122.491 |
| 0.417 | 0.9340 | 7 | -3.908 | 0 | 0.1130 | 0.000278 | 1.50e-03 | 0.1320 | 0.287 | 210.240 | 127.066 |
| 0.537 | 0.7460 | 10 | -5.507 | 0 | 0.1500 | 0.023600 | 1.01e-06 | 0.1560 | 0.252 | 198.267 | 170.062 |
| 0.834 | 0.6240 | 9 | -8.565 | 0 | 0.0838 | 0.072900 | 0.00e+00 | 0.1060 | 0.831 | 232.373 | 115.981 |
| 0.824 | 0.5880 | 6 | -6.400 | 0 | 0.0924 | 0.692000 | 1.04e-04 | 0.1490 | 0.513 | 209.438 | 98.027 |
| 0.838 | 0.5250 | 10 | -3.562 | 1 | 0.0665 | 0.345000 | 1.92e-06 | 0.0771 | 0.884 | 134.256 | 144.981 |
| 0.828 | 0.5210 | 10 | -5.583 | 1 | 0.1370 | 0.858000 | 0.00e+00 | 0.1300 | 0.369 | 200.475 | 106.009 |
| 0.675 | 0.5620 | 7 | -7.678 | 1 | 0.0352 | 0.233000 | 0.00e+00 | 0.0816 | 0.309 | 256.533 | 130.098 |
| 0.685 | 0.7060 | 11 | -7.020 | 0 | 0.0545 | 0.089400 | 2.87e-02 | 0.0986 | 0.640 | 172.293 | 155.989 |
| 0.637 | 0.6430 | 4 | -6.571 | 1 | 0.0519 | 0.130000 | 1.80e-06 | 0.1420 | 0.533 | 200.690 | 97.008 |
| 0.625 | 0.6040 | 10 | -7.415 | 0 | 0.3390 | 0.257000 | 2.06e-05 | 0.1290 | 0.372 | 169.722 | 174.089 |
| 0.518 | 0.5720 | 10 | -6.706 | 0 | 0.0416 | 0.050300 | 2.41e-05 | 0.1290 | 0.291 | 196.767 | 130.053 |
| 0.693 | 0.4940 | 1 | -7.252 | 0 | 0.1080 | 0.139000 | 1.87e-04 | 0.1110 | 0.302 | 241.842 | 135.022 |
| 0.445 | 0.3670 | 0 | -13.086 | 1 | 0.0452 | 0.761000 | 7.50e-02 | 0.0987 | 0.202 | 165.592 | 155.675 |
| 0.610 | 0.5080 | 8 | -6.682 | 0 | 0.1520 | 0.297000 | 0.00e+00 | 0.3840 | 0.758 | 137.876 | 178.818 |
| 0.421 | 0.1310 | 0 | -18.435 | 1 | 0.0382 | 0.952000 | 4.53e-03 | 0.1090 | 0.120 | 291.796 | 137.446 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 0.506 | 0.553 | 10 | -7.751 | 0 | 0.3850 | 6.45e-01 | 0.00e+00 | 0.1220 | 0.3290 | 237.459 | 173.603 |
| 0.699 | 0.668 | 5 | -4.272 | 0 | 0.0336 | 1.54e-01 | 3.20e-06 | 0.3620 | 0.3140 | 148.230 | 144.105 |
| 0.604 | 0.260 | 11 | -10.498 | 1 | 0.0373 | 8.41e-01 | 9.33e-05 | 0.3270 | 0.3180 | 185.689 | 100.901 |
| 0.633 | 0.526 | 6 | -8.433 | 0 | 0.1050 | 4.53e-04 | 1.36e-04 | 0.0800 | 0.2760 | 156.522 | 91.970 |
| 0.485 | 0.627 | 1 | -9.702 | 0 | 0.0416 | 7.37e-02 | 1.14e-05 | 0.3380 | 0.3590 | 198.635 | 92.163 |
| 0.618 | 0.793 | 0 | -5.711 | 1 | 0.0601 | 2.04e-01 | 1.54e-02 | 0.1260 | 0.4590 | 347.526 | 101.015 |
| 0.725 | 0.534 | 11 | -6.238 | 1 | 0.0946 | 7.52e-02 | 0.00e+00 | 0.0919 | 0.5580 | 219.320 | 91.974 |
| 0.735 | 0.795 | 11 | -6.523 | 0 | 0.1130 | 2.96e-02 | 3.18e-05 | 0.0678 | 0.9050 | 188.918 | 122.000 |
| 0.748 | 0.627 | 7 | -6.029 | 1 | 0.0639 | 1.31e-01 | 0.00e+00 | 0.0852 | 0.5240 | 188.491 | 120.963 |
| 0.671 | 0.501 | 4 | -13.119 | 1 | 0.0594 | 3.42e-01 | 0.00e+00 | 0.1940 | 0.8600 | 214.880 | 87.040 |
| 0.664 | 0.771 | 0 | -5.779 | 1 | 0.0533 | 1.55e-02 | 9.60e-06 | 0.4600 | 0.7630 | 265.600 | 109.945 |
| 0.765 | 0.402 | 6 | -6.387 | 1 | 0.0557 | 1.78e-02 | 1.84e-05 | 0.1100 | 0.3530 | 188.854 | 87.054 |
| 0.319 | 0.995 | 0 | -2.940 | 1 | 0.0848 | 6.53e-04 | 0.00e+00 | 0.2740 | 0.3340 | 126.520 | 155.232 |
| 0.515 | 0.479 | 3 | -7.458 | 1 | 0.0261 | 5.44e-01 | 5.98e-03 | 0.1910 | 0.2840 | 209.274 | 88.964 |
| 0.548 | 0.450 | 7 | -7.582 | 1 | 0.0472 | 4.55e-01 | 0.00e+00 | 0.1010 | 0.3290 | 161.290 | 185.960 |
| 0.766 | 0.622 | 5 | -5.292 | 0 | 0.1010 | 6.39e-02 | 0.00e+00 | 0.2070 | 0.6790 | 191.493 | 140.076 |
| 0.490 | 0.386 | 8 | -6.160 | 1 | 0.0357 | 7.52e-01 | 0.00e+00 | 0.1040 | 0.2330 | 221.824 | 80.599 |
| 0.180 | 0.934 | 4 | -8.699 | 1 | 0.1150 | 3.32e-05 | 5.51e-04 | 0.0702 | 0.4330 | 130.267 | 197.043 |
| 0.583 | 0.621 | 5 | -6.902 | 0 | 0.0479 | 9.72e-03 | 1.06e-03 | 0.1810 | 0.3990 | 323.480 | 140.036 |
| 0.896 | 0.675 | 1 | -3.908 | 1 | 0.1240 | 3.46e-01 | 8.11e-06 | 0.2620 | 0.6590 | 213.989 | 131.999 |
| 0.785 | 0.871 | 4 | -2.692 | 0 | 0.2650 | 1.44e-01 | 0.00e+00 | 0.3090 | 0.3150 | 290.427 | 87.248 |
| 0.644 | 0.211 | 0 | -13.966 | 1 | 0.0394 | 8.98e-01 | 9.70e-02 | 0.0757 | 0.4070 | 169.045 | 140.058 |
| 0.391 | 0.205 | 9 | -14.148 | 1 | 0.0452 | 7.63e-01 | 2.47e-03 | 0.3640 | 0.1140 | 315.716 | 78.004 |
| 0.502 | 0.898 | 2 | -8.912 | 1 | 0.0469 | 1.64e-03 | 0.00e+00 | 0.3360 | 0.7650 | 120.333 | 117.246 |

| Danceability | Energy | Key | Loudness | Mode | Speechiness | Acousticness | Instrumentalness | Liveness | Valence | Duration | Tempo |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.646 | 0.770 | 2 | -6.596 | 0 | 0.2260 | 2.49e-03 | 0.00e+00 | 0.0715 | 0.6810 | 236.133 | 99.165 |
| 0.681 | 0.514 | 1 | -6.272 | 0 | 0.0676 | 1.91e-01 | 2.07e-02 | 0.0983 | 0.1450 | 194.120 | 150.979 |
| 0.754 | 0.702 | 10 | -6.378 | 0 | 0.2640 | 3.12e-01 | 0.00e+00 | 0.2890 | 0.3830 | 230.800 | 145.959 |
| 0.666 | 0.542 | 8 | -6.429 | 1 | 0.0392 | 2.70e-01 | 0.00e+00 | 0.0765 | 0.0771 | 228.787 | 120.134 |
| 0.507 | 0.790 | 0 | -7.307 | 0 | 0.0294 | 2.33e-01 | 1.39e-01 | 0.1450 | 0.6000 | 320.467 | 147.065 |
| 0.667 | 0.659 | 2 | -4.932 | 1 | 0.0298 | 1.13e-01 | 0.00e+00 | 0.3250 | 0.5750 | 203.440 | 89.128 |

On transforme l'output Liked en 0 et 1

In [48]:
```
library(plyr)
Y1 = (revalue(don1$Liked, c("Yes"=1, "No"=0)))
Y = as.numeric(matrix(Y1))
Y
```

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 0 0 1 0 0 0 1 1 1 0 0 1 1 0 1 1 0 1 1 0 0 1 1 0 0 0 1 1 1 1 0
0 0 1 0 1 1 1 0 1 1 1 1 0 1 0 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 1 0 1 1 0 1 1 1 0 1 1 0 1 1 1
0 1 1 1 1 0 1 1 0 1 1 1 1 0 1 1 0 1 1 0 1 1 0 1 1 0
```

In [49]:

```
#Y est la variable Liked codée en 0 ou 1
don$Y = Y
don
```

| Danceability | Energy | Key | Loudness | Mode | Speechiness | Acousticness | Instrumentalness | Liveness | Valence | Duration | Tempo | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.502 | 0.6400 | 6 | -9.702 | 0 | 0.0286 | 0.031300 | 2.16e-06 | 0.1540 | 0.449 | 210.730 | 131.551 | 1 |
| 0.749 | 0.4910 | 10 | -9.655 | 1 | 0.0403 | 0.020000 | 8.99e-03 | 0.1590 | 0.536 | 292.613 | 119.528 | 1 |
| 0.664 | 0.6090 | 1 | -6.509 | 1 | 0.0707 | 0.304000 | 0.00e+00 | 0.0926 | 0.194 | 210.560 | 130.041 | 1 |
| 0.805 | 0.4980 | 7 | -7.927 | 0 | 0.0737 | 0.020300 | 2.37e-05 | 0.0850 | 0.636 | 214.215 | 121.006 | 1 |
| 0.645 | 0.5340 | 6 | -10.800 | 0 | 0.0479 | 0.157000 | 0.00e+00 | 0.0863 | 0.463 | 226.467 | 105.020 | 1 |
| 0.365 | 0.2730 | 4 | -16.526 | 0 | 0.0380 | 0.940000 | 4.31e-01 | 0.1090 | 0.238 | 248.695 | 132.285 | 1 |
| 0.778 | 0.6950 | 4 | -6.865 | 0 | 0.0913 | 0.175000 | 0.00e+00 | 0.1500 | 0.472 | 132.780 | 149.996 | 1 |
| 0.782 | 0.5590 | 6 | -7.106 | 0 | 0.0767 | 0.125000 | 0.00e+00 | 0.3850 | 0.685 | 188.344 | 129.992 | 1 |
| 0.509 | 0.5440 | 5 | -6.798 | 0 | 0.0307 | 0.630000 | 1.48e-01 | 0.1000 | 0.206 | 241.747 | 71.502 | 1 |
| 0.743 | 0.6220 | 4 | -5.332 | 0 | 0.1360 | 0.904000 | 1.66e-05 | 0.1430 | 0.317 | 183.133 | 140.054 | 1 |
| 0.501 | 0.0958 | 0 | -15.605 | 1 | 0.0433 | 0.770000 | 0.00e+00 | 0.2980 | 0.204 | 248.808 | 108.741 | 1 |
| 0.653 | 0.7020 | 4 | -10.106 | 0 | 0.1170 | 0.023200 | 3.36e-02 | 0.0864 | 0.258 | 211.559 | 169.950 | 1 |
| 0.762 | 0.7290 | 6 | -8.076 | 0 | 0.0679 | 0.481000 | 1.37e-04 | 0.1080 | 0.482 | 225.040 | 119.956 | 1 |
| 0.657 | 0.6430 | 8 | -6.661 | 0 | 0.0433 | 0.482000 | 2.08e-06 | 0.1890 | 0.394 | 251.670 | 82.516 | 1 |
| 0.845 | 0.7170 | 5 | -6.771 | 0 | 0.2110 | 0.666000 | 0.00e+00 | 0.1100 | 0.686 | 160.893 | 122.491 | 0 |
| 0.417 | 0.9340 | 7 | -3.908 | 0 | 0.1130 | 0.000278 | 1.50e-03 | 0.1320 | 0.287 | 210.240 | 127.066 | 0 |
| 0.537 | 0.7460 | 10 | -5.507 | 0 | 0.1500 | 0.023600 | 1.01e-06 | 0.1560 | 0.252 | 198.267 | 170.062 | 1 |
| 0.834 | 0.6240 | 9 | -8.565 | 0 | 0.0838 | 0.072900 | 0.00e+00 | 0.1060 | 0.831 | 232.373 | 115.981 | 0 |
| 0.824 | 0.5880 | 6 | -6.400 | 0 | 0.0924 | 0.692000 | 1.04e-04 | 0.1490 | 0.513 | 209.438 | 98.027 | 0 |
| 0.838 | 0.5250 | 10 | -3.562 | 1 | 0.0665 | 0.345000 | 1.92e-06 | 0.0771 | 0.884 | 134.256 | 144.981 | 0 |
| 0.828 | 0.5210 | 10 | -5.583 | 1 | 0.1370 | 0.858000 | 0.00e+00 | 0.1300 | 0.369 | 200.475 | 106.009 | 1 |
| 0.675 | 0.5620 | 7 | -7.678 | 1 | 0.0352 | 0.233000 | 0.00e+00 | 0.0816 | 0.309 | 256.533 | 130.098 | 1 |
| 0.685 | 0.7060 | 11 | -7.020 | 0 | 0.0545 | 0.089400 | 2.87e-02 | 0.0986 | 0.640 | 172.293 | 155.989 | 1 |
| 0.637 | 0.6430 | 4 | -6.571 | 1 | 0.0519 | 0.130000 | 1.80e-06 | 0.1420 | 0.533 | 200.690 | 97.008 | 0 |
| 0.625 | 0.6040 | 10 | -7.415 | 0 | 0.3390 | 0.257000 | 2.06e-05 | 0.1290 | 0.372 | 169.722 | 174.089 | 0 |
| 0.518 | 0.5720 | 10 | -6.706 | 0 | 0.0416 | 0.050300 | 2.41e-05 | 0.1290 | 0.291 | 196.767 | 130.053 | 1 |
| 0.693 | 0.4940 | 1 | -7.252 | 0 | 0.1080 | 0.139000 | 1.87e-04 | 0.1110 | 0.302 | 241.842 | 135.022 | 1 |
| 0.445 | 0.3670 | 0 | -13.086 | 1 | 0.0452 | 0.761000 | 7.50e-02 | 0.0987 | 0.202 | 165.592 | 155.675 | 0 |
| 0.610 | 0.5080 | 8 | -6.682 | 0 | 0.1520 | 0.297000 | 0.00e+00 | 0.3840 | 0.758 | 137.876 | 178.818 | 1 |
| 0.421 | 0.1310 | 0 | -18.435 | 1 | 0.0382 | 0.952000 | 4.53e-03 | 0.1090 | 0.120 | 291.796 | 137.446 | 1 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 0.506 | 0.553 | 10 | -7.751 | 0 | 0.3850 | 6.45e-01 | 0.00e+00 | 0.1220 | 0.3290 | 237.459 | 173.603 | 0 |
| 0.699 | 0.668 | 5 | -4.272 | 0 | 0.0336 | 1.54e-01 | 3.20e-06 | 0.3620 | 0.3140 | 148.230 | 144.105 | 1 |
| 0.604 | 0.260 | 11 | -10.498 | 1 | 0.0373 | 8.41e-01 | 9.33e-05 | 0.3270 | 0.3180 | 185.689 | 100.901 | 1 |
| 0.633 | 0.526 | 6 | -8.433 | 0 | 0.1050 | 4.53e-04 | 1.36e-04 | 0.0800 | 0.2760 | 156.522 | 91.970 | 1 |
| 0.485 | 0.627 | 1 | -9.702 | 0 | 0.0416 | 7.37e-02 | 1.14e-05 | 0.3380 | 0.3590 | 198.635 | 92.163 | 0 |
| 0.618 | 0.793 | 0 | -5.711 | 1 | 0.0601 | 2.04e-01 | 1.54e-02 | 0.1260 | 0.4590 | 347.526 | 101.015 | 1 |
| 0.725 | 0.534 | 11 | -6.238 | 1 | 0.0946 | 7.52e-02 | 0.00e+00 | 0.0919 | 0.5580 | 219.320 | 91.974 | 1 |
| 0.735 | 0.795 | 11 | -6.523 | 0 | 0.1130 | 2.96e-02 | 3.18e-05 | 0.0678 | 0.9050 | 188.918 | 122.000 | 1 |
| 0.748 | 0.627 | 7 | -6.029 | 1 | 0.0639 | 1.31e-01 | 0.00e+00 | 0.0852 | 0.5240 | 188.491 | 120.963 | 1 |
| 0.671 | 0.501 | 4 | -13.119 | 1 | 0.0594 | 3.42e-01 | 0.00e+00 | 0.1940 | 0.8600 | 214.880 | 87.040 | 0 |
| 0.664 | 0.771 | 0 | -5.779 | 1 | 0.0533 | 1.55e-02 | 9.60e-06 | 0.4600 | 0.7630 | 265.600 | 109.945 | 1 |
| 0.765 | 0.402 | 6 | -6.387 | 1 | 0.0557 | 1.78e-02 | 1.84e-05 | 0.1100 | 0.3530 | 188.854 | 87.054 | 1 |
| 0.319 | 0.995 | 0 | -2.940 | 1 | 0.0848 | 6.53e-04 | 0.00e+00 | 0.2740 | 0.3340 | 126.520 | 155.232 | 0 |
| 0.515 | 0.479 | 3 | -7.458 | 1 | 0.0261 | 5.44e-01 | 5.98e-03 | 0.1910 | 0.2840 | 209.274 | 88.964 | 1 |
| 0.548 | 0.450 | 7 | -7.582 | 1 | 0.0472 | 4.55e-01 | 0.00e+00 | 0.1010 | 0.3290 | 161.290 | 185.960 | 1 |
| 0.766 | 0.622 | 5 | -5.292 | 0 | 0.1010 | 6.39e-02 | 0.00e+00 | 0.2070 | 0.6790 | 191.493 | 140.076 | 1 |
| 0.490 | 0.386 | 8 | -6.160 | 1 | 0.0357 | 7.52e-01 | 0.00e+00 | 0.1040 | 0.2330 | 221.824 | 80.599 | 1 |
| 0.180 | 0.934 | 4 | -8.699 | 1 | 0.1150 | 3.32e-05 | 5.51e-04 | 0.0702 | 0.4330 | 130.267 | 197.043 | 0 |
| 0.583 | 0.621 | 5 | -6.902 | 0 | 0.0479 | 9.72e-03 | 1.06e-03 | 0.1810 | 0.3990 | 323.480 | 140.036 | 1 |
| 0.896 | 0.675 | 1 | -3.908 | 1 | 0.1240 | 3.46e-01 | 8.11e-06 | 0.2620 | 0.6590 | 213.989 | 131.999 | 1 |
| 0.785 | 0.871 | 4 | -2.692 | 0 | 0.2650 | 1.44e-01 | 0.00e+00 | 0.3090 | 0.3150 | 290.427 | 87.248 | 0 |
| 0.644 | 0.211 | 0 | -13.966 | 1 | 0.0394 | 8.98e-01 | 9.70e-02 | 0.0757 | 0.4070 | 169.045 | 140.058 | 1 |
| 0.391 | 0.205 | 9 | -14.148 | 1 | 0.0452 | 7.63e-01 | 2.47e-03 | 0.3640 | 0.1140 | 315.716 | 78.004 | 1 |
| 0.502 | 0.898 | 2 | -8.912 | 1 | 0.0469 | 1.64e-03 | 0.00e+00 | 0.3360 | 0.7650 | 120.333 | 117.246 | 0 |

| Danceability | Energy | Key | Loudness | Mode | Speechiness | Acousticness | Instrumentalness | Liveness | Valence | Duration | Tempo | Y |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.646 | 0.770 | 2 | -6.596 | 0 | 0.2260 | 2.49e-03 | 0.00e+00 | 0.0715 | 0.6810 | 236.133 | 99.165 | 1 |
| 0.681 | 0.514 | 1 | -6.272 | 0 | 0.0676 | 1.91e-01 | 2.07e-02 | 0.0983 | 0.1450 | 194.120 | 150.979 | 1 |
| 0.754 | 0.702 | 10 | -6.378 | 0 | 0.2640 | 3.12e-01 | 0.00e+00 | 0.2890 | 0.3830 | 230.800 | 145.959 | 0 |
| 0.666 | 0.542 | 8 | -6.429 | 1 | 0.0392 | 2.70e-01 | 0.00e+00 | 0.0765 | 0.0771 | 228.787 | 120.134 | 1 |
| 0.507 | 0.790 | 0 | -7.307 | 0 | 0.0294 | 2.33e-01 | 1.39e-01 | 0.1450 | 0.6000 | 320.467 | 147.065 | 1 |
| 0.667 | 0.659 | 2 | -4.932 | 1 | 0.0298 | 1.13e-01 | 0.00e+00 | 0.3250 | 0.5750 | 203.440 | 89.128 | 0 |

Regression multiple afin de pouvoir tester une méthode par la suite :

In [8]:
```
fit.lm = lm(Y~., data=don)
summary(fit.lm)
```

```

Call:
lm(formula = Y ~ ., data = don)

Residuals:
    Min      1Q  Median      3Q     Max 
-0.8823 -0.3050  0.1146  0.2604  0.7412 

Coefficients:
                   Estimate Std. Error t value Pr(>|t|)   
(Intercept)       1.375e+00  4.833e-01   2.844  0.00537 **
Danceability      7.516e-01  3.302e-01   2.276  0.02491 * 
Energy           -6.739e-01  3.772e-01  -1.786  0.07697 . 
Key               5.717e-03  1.115e-02   0.513  0.60917   
Loudness          1.710e-02  2.056e-02   0.832  0.40749   
Mode             -1.131e-01  8.532e-02  -1.325  0.18807   
Speechiness      -1.760e+00  6.056e-01  -2.906  0.00448 **
Acousticness     -1.845e-01  1.867e-01  -0.989  0.32517   
Instrumentalness  1.415e-01  2.982e-01   0.475  0.63611   
Liveness         -5.129e-01  2.860e-01  -1.793  0.07587 . 
Valence          -7.207e-01  2.203e-01  -3.271  0.00146 **
Duration         -2.980e-06  2.022e-06  -1.474  0.14364   
Tempo            -1.998e-06  1.826e-06  -1.094  0.27663   
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 0.4127 on 103 degrees of freedom
Multiple R-squared:  0.3037,	Adjusted R-squared:  0.2226 
F-statistic: 3.744 on 12 and 103 DF,  p-value: 0.0001045
```

Méthode 0 : Utilisation de la fonction Step qui se base sur l'AIC

In [9]:
```
step(fit.lm)
```

```
               Df Sum of Sq    RSS     AIC
<none>                      18.054 -199.79
- Mode          1   0.32511 18.379 -199.72
- Energy        1   0.33471 18.389 -199.66
- Duration      1   0.33790 18.392 -199.64
- Liveness      1   0.72818 18.782 -197.20
- Danceability  1   1.43819 19.492 -192.90
- Speechiness   1   1.61784 19.672 -191.83
- Valence       1   2.07747 20.131 -189.15


Call:
lm(formula = Y ~ Danceability + Energy + Mode + Speechiness + 
    Liveness + Valence + Duration, data = don)

Coefficients:
 (Intercept)  Danceability        Energy          Mode   Speechiness  
   9.493e-01     8.903e-01    -3.179e-01    -1.168e-01    -1.843e+00  
    Liveness       Valence      Duration  
  -5.754e-01    -7.371e-01    -2.816e-06  
```

In [10]:
```
library(leaps)
```

```
Warning message:
"package 'leaps' was built under R version 3.6.3"
```

Méthode 1 : Choix de meilleurs sous-ensembles

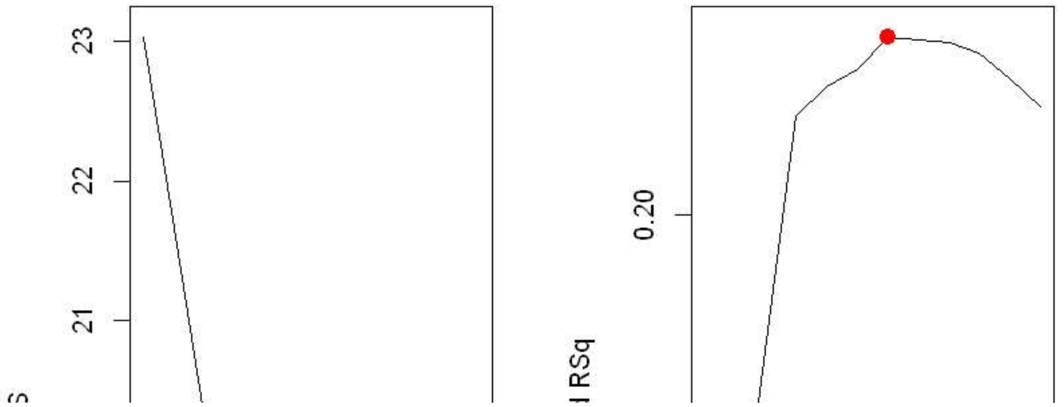
In [11]:
```
reg.mod=regsubsets(Y~., data=don, nvmax=12)
reg.sum= summary(reg.mod)
reg.sum
```

```
Subset selection object
Call: regsubsets.formula(Y ~ ., data = don, nvmax = 12)
12 Variables  (and intercept)
                 Forced in Forced out
Danceability         FALSE      FALSE
Energy               FALSE      FALSE
Key                  FALSE      FALSE
Loudness             FALSE      FALSE
Mode                 FALSE      FALSE
Speechiness          FALSE      FALSE
Acousticness         FALSE      FALSE
Instrumentalness     FALSE      FALSE
Liveness             FALSE      FALSE
Valence              FALSE      FALSE
Duration             FALSE      FALSE
Tempo                FALSE      FALSE
1 subsets of each size up to 12
Selection Algorithm: exhaustive
          Danceability Energy Key Loudness Mode Speechiness Acousticness
1  ( 1 )  " "          "*"    " " " "      " "  " "         " "
2  ( 1 )  " "          " "    " " " "      " "  " "         " "
3  ( 1 )  "*"          " "    " " " "      " "  "*"         " "
4  ( 1 )  "*"          " "    " " " "      " "  "*"         " "
5  ( 1 )  "*"          " "    " " " "      " "  "*"         " "
6  ( 1 )  "*"          "*"    " " " "      " "  "*"         " "
7  ( 1 )  "*"          "*"    " " " "      "*"  "*"         " "
8  ( 1 )  "*"          "*"    " " " "      "*"  "*"         " "
9  ( 1 )  "*"          "*"    " " " "      "*"  "*"         "*"
10  ( 1 ) "*"          "*"    " " "*"      "*"  "*"         "*"
11  ( 1 ) "*"          "*"    "*" "*"      "*"  "*"         "*"
12  ( 1 ) "*"          "*"    "*" "*"      "*"  "*"         "*"
          Instrumentalness Liveness Valence Duration Tempo
1  ( 1 )  " "              " "      " "     " "      " "
2  ( 1 )  " "              "*"      "*"     " "      " "
3  ( 1 )  " "              " "      "*"     " "      " "
4  ( 1 )  " "              "*"      "*"     " "      " "
5  ( 1 )  " "              "*"      "*"     "*"      " "
6  ( 1 )  " "              "*"      "*"     "*"      " "
7  ( 1 )  " "              "*"      "*"     "*"      " "
8  ( 1 )  " "              "*"      "*"     "*"      "*"
9  ( 1 )  " "              "*"      "*"     "*"      "*"
10  ( 1 ) " "              "*"      "*"     "*"      "*"
11  ( 1 ) " "              "*"      "*"     "*"      "*"
12  ( 1 ) "*"              "*"      "*"     "*"      "*"
```

In [12]:
```
which.max(reg.sum$adjr2)
```

```
7
```

In [13]:
```
par(mfrow=c(1,2))
plot(reg.sum$rss,xlab="Number of Variables",ylab="RSS",type="l")
plot(reg.sum$adjr2,xlab="Number of Variables",ylab="Adjusted RSq",type="l")
points(7,reg.sum$adjr2[7], col="red",cex=2,pch=20)
```



In [104]:
```
coef(reg.mod,7)
```

```
 (Intercept)   0.94930854265269
Danceability   0.890278781821163
      Energy   -0.317889103744736
        Mode   -0.116798447920091
 Speechiness   -1.84286440239108
    Liveness   -0.575396979295119
     Valence   -0.737075868494187
    Duration   -2.81552562161636e-06
```

In [14]:
```
set.seed(2226947)
#valeurs1=matrix(data=c("Valeur K", "Taux d'erreur"), ncol=2, byrow = TRUE)
for (j in 1:4){
    train_data <- don[-((j*29 -28):(j*29)), ]
    test_data <- don[((j*29 -28):(j*29)), ]

    regsub= regsubsets(Y~., data = train_data, nvmax=12)
    summary(regsub)
}
```

In [15]:
```
k = 4       # number of folds
set.seed(2226947)

folds = sample(1:k, nrow(don), replace = TRUE)

err = matrix(NA, k, 12, dimnames = list(NULL, paste(1:12)))
```

In [16]:
```
predict.regsubsets = function(object,newdata,id,...){
      form = as.formula(object$call[[2]])
      mat = model.matrix(form,newdata)
      coefi = coef(object,id=id)
      xvars = names(coefi)
      mat[,xvars]%*%coefi
}
```

In [17]:
```
# Outer loop iterates over all folds
for(j in 1:k){

    # The perform best subset selection on the full dataset, minus the jth fold
    best_fit = regsubsets(Y~., data = don[folds!=j,], nvmax=12)

    # Inner loop iterates over each size i
    for(i in 1:12){

        # Predict the values of the current fold from the "best subset" model on i predictors
        pred = predict(best_fit, don[folds==j,], id=i)

        # Calculate the MSE, store it in the matrix we created above
        err[j,i] = mean((don$Y[folds==j]-pred)^2)
    }
}
```

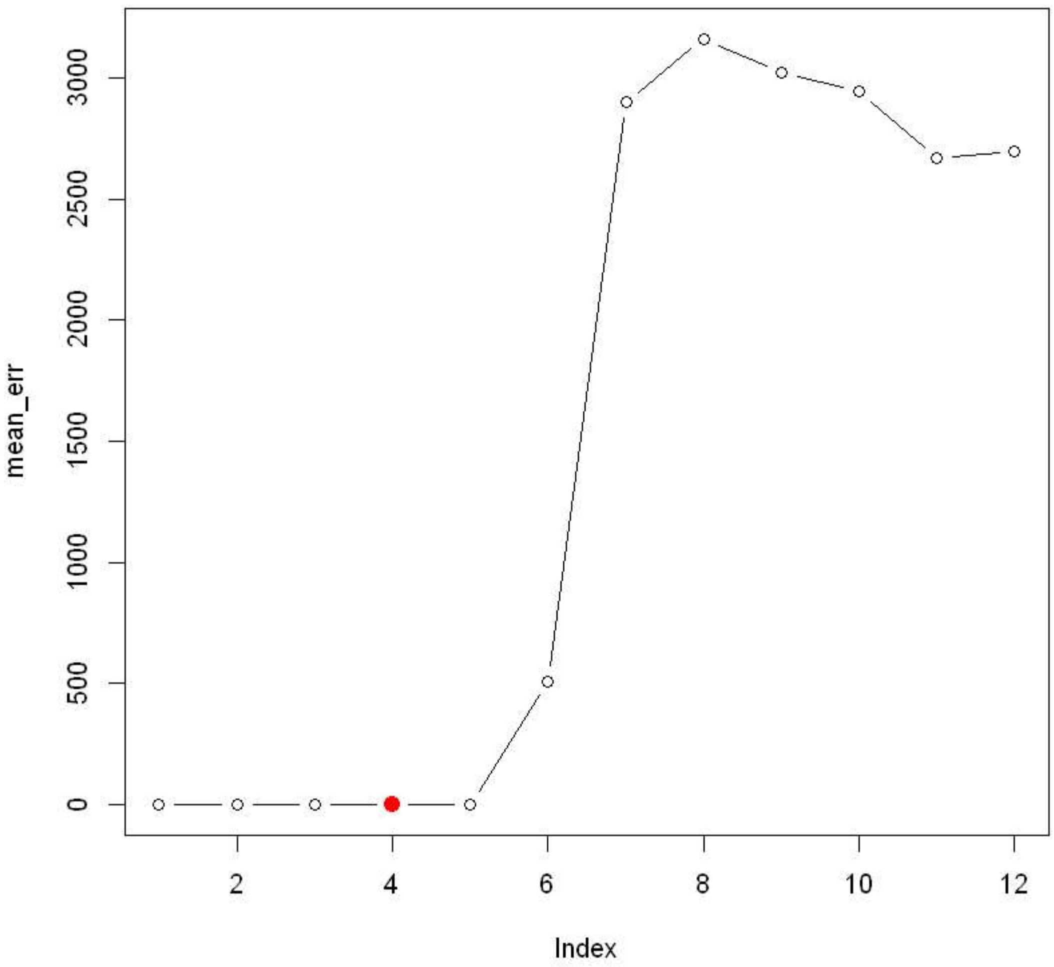
In [18]:
```
# Take the mean of over all folds for each model size
mean_err = apply(err, 2, mean)
err
mean_err


# Find the model size with the smallest cross-validation error
min = which.min(mean_err)

# Plot the cross-validation error for each model size, highlight the min
plot(mean_err, type='b')
points(min, mean_err[min][1], col = "red", cex = 2, pch = 20)
```

| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0.2636625 | 0.2623522 | 0.2380631 | 0.2409974 | 0.2431175 | 0.2477781 | 0.2469477 | 2.394767e-01 | 0.2490127 | 0.2552201 | 0.2377066 | 0.2386266 |
| 0.2319110 | 0.2421341 | 0.2204401 | 0.2226099 | 0.2276376 | 0.2493920 | 0.2648501 | 2.634830e-01 | 0.2177543 | 0.2142297 | 0.2142856 | 0.2146316 |
| 0.1885998 | 0.1892376 | 0.1819801 | 0.1715273 | 0.1854597 | 2030.6439699 | 2046.0261804 | 2.574892e+03 | 2238.9425856 | 2159.8729943 | 2187.2199480 | 2240.9495701 |
| 0.2088486 | 0.2177904 | 0.2043900 | 0.1895745 | 0.1911745 | 0.1912477 | 9552.0751883 | 1.006546e+04 | 9853.6208938 | 9609.7455420 | 8496.8821390 | 8539.9992016 |

| | |
|---|---|
| **1** | 0.223255479090055 |
| **2** | 0.227878589570813 |
| **3** | 0.211218336716169 |
| **4** | 0.206177270259003 |
| **5** | 0.211847317739552 |
| **6** | 507.833096919807 |
| **7** | 2899.65329162691 |
| **8** | 3160.21375103754 |
| **9** | 3023.25756159433 |
| **10** | 2942.52199651893 |
| **11** | 2671.1385197973 |
| **12** | 2695.35050748879 |



In [19]:
```
reg_best = regsubsets(Y~., data = don, nvmax = 12)
coef(reg_best, 4)
```

```
 (Intercept)   0.712607038565121
Danceability   0.971118873548491
 Speechiness   -1.71083426848013
    Liveness   -0.745126085550979
     Valence   -0.844130615704666
```

In [20]:
```
library(glmnet)
```

```
Warning message:
"package 'glmnet' was built under R version 3.6.3"Loading required package: Matrix
Warning message:
"package 'Matrix' was built under R version 3.6.3"Loaded glmnet 4.1-1
```

Méthode 2 : Lasso

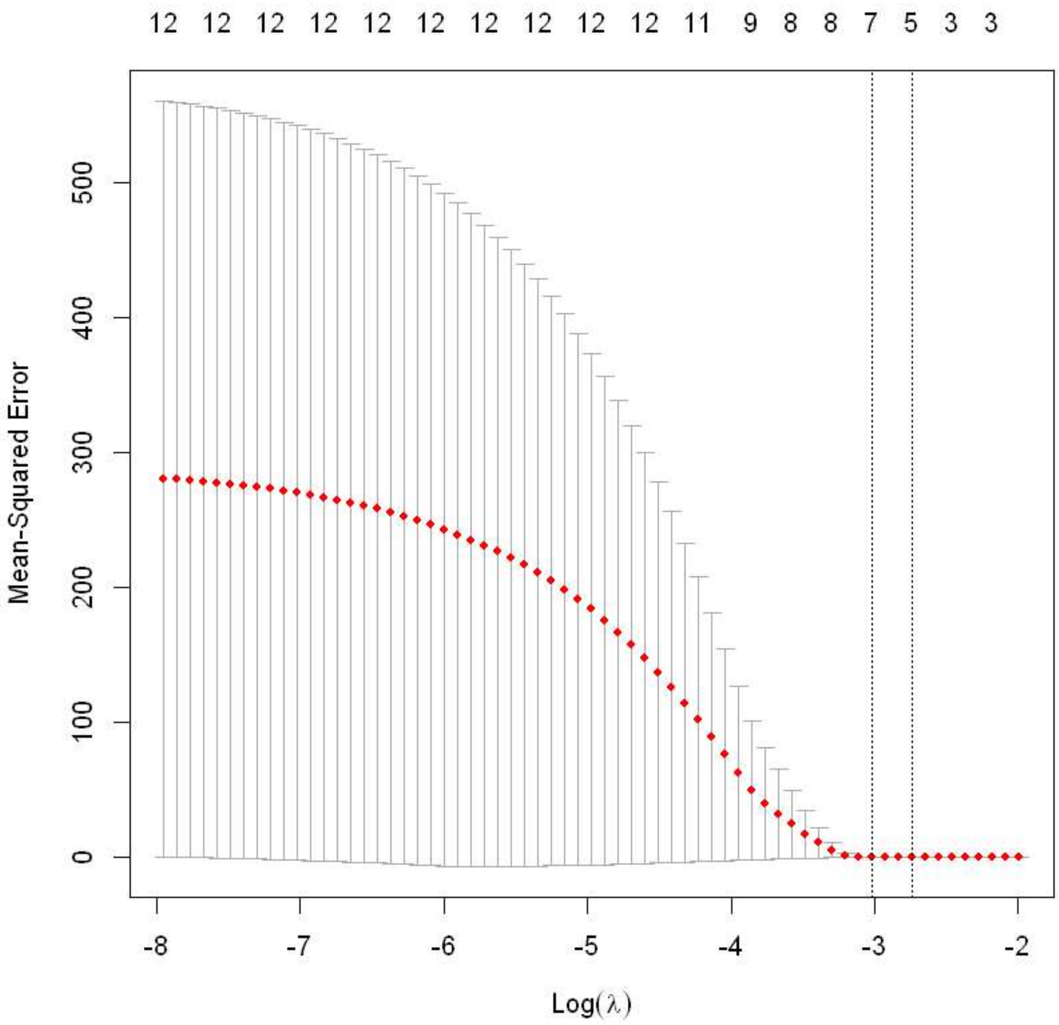
In [23]:
```
#Lasso :

grid<-10^seq(10,-2,length=100)
```

In [24]:
```
x<-model.matrix(Y~.,don)
```

In [25]:
```
lasso.mod<-glmnet(x,Y,alpha=1,lambda=grid)
```

```
In [26]: cv_out=cv.glmnet(x,Y,alpha=1)
         plot(cv_out)
```



```
In [27]: bestlam<-cv_out$lambda.min # la valeur de lambda pour laquelle MSE est min
         bestlam
```

```
0.0490970546599512
```

```
In [32]: set.seed(2226947)
         #valeurs1=matrix(data=c("Valeur K", "Taux d'erreur"), ncol=2, byrow = TRUE)
         mse = rep(0,4)
         for (j in 1:4){
             train_data_1 <- don[-((j*29-28):(j*29)), -13]
             result_data1 = don[-((j*29-28):(j*29)), 13]
             test_data_1 <- don[((j*29-28):(j*29)), -13]
             test_data1 <- don[((j*29-28):(j*29)), 13]


             lasso.mod<-glmnet(train_data_1,result_data1,alpha=1,lambda=grid)
             lasso.pred=predict(lasso.mod,s=bestlam,newx=as.matrix(test_data_1)
         )
             mse[j]=mean((lasso.pred-test_data1)^2)

         }
         mse
         mean(mse)
```

```
0.188687702447149    77.3451253953816    0.202030558423769    0.172628834125863

19.4771181225946
```

In [33]:

```
out=glmnet(x,Y,alpha=1,lambda=grid)
lasso.coef=predict(out,type="coefficients",s=bestlam)[1:13,]
lasso.coef
lasso.coef[lasso.coef!=0]
```

```
     (Intercept)   0.92748592906131
     (Intercept)   0
    Danceability   0.271120821289483
          Energy   -0.235840801289313
             Key   0
        Loudness   0
            Mode   -0.00127613658312779
     Speechiness   -0.581987728471468
    Acousticness   0
Instrumentalness   0
        Liveness   -0.408923572338646
         Valence   -0.352546883641634
        Duration   -1.7719889127453e-07

     (Intercept)   0.92748592906131
    Danceability   0.271120821289483
          Energy   -0.235840801289313
            Mode   -0.00127613658312779
     Speechiness   -0.581987728471468
        Liveness   -0.408923572338646
         Valence   -0.352546883641634
        Duration   -1.7719889127453e-07
```

In [40]:

```
don_stdv <-scale(don[,-13])
don_std <- data.frame(don_stdv)

don_std$Y=Y
str(don_std)
```

```
'data.frame':	116 obs. of  13 variables:
 $ Danceability    : num  -0.9567 0.7342 0.1523 1.1175 0.0222 ...
 $ Energy          : num  0.2492 -0.5264 0.0878 -0.49 -0.3026 ...
 $ Key             : num  0.134 1.222 -1.227 0.406 0.134 ...
 $ Loudness        : num  -0.693 -0.677 0.363 -0.106 -1.056 ...
 $ Mode            : num  -0.979 1.013 1.013 -0.979 -0.979 ...
 $ Speechiness     : num  -0.823 -0.655 -0.218 -0.175 -0.545 ...
 $ Acousticness    : num  -0.94266 -0.98193 0.00489 -0.98089 -0.5059 ...
 $ Instrumentalness: num  -0.3 -0.237 -0.3 -0.3 -0.3 ...
 $ Liveness        : num  -0.171 -0.136 -0.594 -0.647 -0.638 ...
 $ Valence         : num  0.0232 0.4199 -1.1397 0.8759 0.087 ...
 $ Duration        : num  -0.0929 -0.0887 -0.0929 -0.0927 -0.0921 ...
 $ Tempo           : num  -0.0925 -0.0931 -0.0926 -0.093 -0.0937 ...
 $ Y               : num  1 1 1 1 1 1 1 1 1 1 ...
```

In [42]:

```
library(class)
```

```
Warning message:
"package 'class' was built under R version 3.6.3"
```

Méthode 3 : KNN

In [46]:

```
#KNN - 4-FOLD
K=50
#Recherche du k optimal :
set.seed(2226947)
valeurs1=matrix(data=c("Valeur K", "Taux d'erreur"), ncol=2, byrow = TRUE)
taux_err1 <- rep(0,4)

for(i in 1:K){
for (j in 1:4){
train_data2 <- don_std[-((j*29-28):(j*29)), -13]
test_data2 <- don_std[((j*29-28):(j*29)), -13]
class_entr2 <- don[-((j*29 -28):(j*29)), 13]
class_test2 <- don[((j*29-28):(j*29)), 13]
knn_fold = knn(train=train_data2, test=test_data2, cl=class_entr2,k=i, prob=TRUE)
taux_err1[j] = mean(knn_fold != don[((j*29-28):(j*29)), 13])
                 }
    valeurs1 = rbind(valeurs1, c(i,mean(taux_err1)))
}

valeurs1
valeurs1[8,] #taux d'erreur de K=7
min(valeurs1)
```

| Valeur K | Taux d'erreur |
|---:|---:|
| 1 | 0.310344827586207 |
| 2 | 0.387931034482759 |
| 3 | 0.267241379310345 |
| 4 | 0.293103448275862 |
| 5 | 0.28448275862069 |
| 6 | 0.318965517241379 |
| 7 | 0.258620689655172 |
| 8 | 0.301724137931034 |
| 9 | 0.28448275862069 |
| 10 | 0.28448275862069 |
| 11 | 0.293103448275862 |
| 12 | 0.267241379310345 |
| 13 | 0.327586206896552 |
| 14 | 0.301724137931034 |
| 15 | 0.310344827586207 |
| 16 | 0.318965517241379 |
| 17 | 0.327586206896552 |
| 18 | 0.318965517241379 |
| 19 | 0.318965517241379 |
| 20 | 0.318965517241379 |
| 21 | 0.310344827586207 |
| 22 | 0.310344827586207 |
| 23 | 0.318965517241379 |
| 24 | 0.318965517241379 |
| 25 | 0.318965517241379 |
| 26 | 0.318965517241379 |
| 27 | 0.318965517241379 |
| 28 | 0.318965517241379 |
| 29 | 0.318965517241379 |
| 30 | 0.318965517241379 |
| 31 | 0.318965517241379 |
| 32 | 0.318965517241379 |
| 33 | 0.318965517241379 |
| 34 | 0.318965517241379 |
| 35 | 0.318965517241379 |
| 36 | 0.318965517241379 |
| 37 | 0.318965517241379 |
| 38 | 0.318965517241379 |
| 39 | 0.318965517241379 |
| 40 | 0.318965517241379 |
| 41 | 0.318965517241379 |
| 42 | 0.318965517241379 |
| 43 | 0.318965517241379 |
| 44 | 0.318965517241379 |
| 45 | 0.318965517241379 |
| 46 | 0.318965517241379 |
| 47 | 0.318965517241379 |
| 48 | 0.318965517241379 |
| 49 | 0.318965517241379 |
| 50 | 0.318965517241379 |

```
'7'    '0.258620689655172'
```

```
'0.258620689655172'
```

In [101]:
```
# KNN 4-FOLD AVEC LA MEILLEURE VALEUR DE K =7
valeurs7=matrix(data=c("Valeur K", "Taux d'erreur"), ncol=2, byrow = TRUE)
taux_err7 <- rep(0,4)


for (j in 1:4){
train_data7 <- don_std[-((j*29-28):(j*29)), -13]
test_data7<- don_std[((j*29-28):(j*29)), -13]
class_entr7 <- don[-((j*29 -28):(j*29)), 13]
class_test7 <- don[((j*29-28):(j*29)), 13]
knn_fold7 = knn(train=train_data7, test=test_data7, cl=class_entr7,k=7, prob=TRUE)
taux_err7[j] = mean(knn_fold7 != class_test7)
    print(table(class_test7, knn_fold7))
                 }
    valeurs7 = rbind(valeurs7, c(i,mean(taux_err7)))
```

```
           knn_fold7
class_test7  0  1
          0  1  7
          1  0 21
           knn_fold7
class_test7  0  1
          0  5  7
          1  0 17
           knn_fold7
class_test7  0  1
          0  2  7
          1  2 18
           knn_fold7
class_test7  0  1
          0  3  5
          1  2 19
```

In [79]:
```
don_entr<-don[-(20:67),]

don_test<-don[20:67,]
```

Méthode 4 : Régression logsitique

In [80]:
```
fit.logb<-glm(Y~., family=binomial,data=don_entr)
summary(fit.logb)
```

```

Call:
glm(formula = Y ~ ., family = binomial, data = don_entr)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.9972  -0.2461   0.2480   0.5591   1.9137

Coefficients:
                   Estimate Std. Error z value Pr(>|z|)
(Intercept)        2.213919   6.082697   0.364   0.7159
Danceability       7.796753   3.690739   2.113   0.0346 *
Energy            -8.224604   4.383401  -1.876   0.0606 .
Key               -0.031063   0.105983  -0.293   0.7694
Loudness           0.287478   0.234606   1.225   0.2204
Mode              -0.094241   0.996176  -0.095   0.9246
Speechiness      -12.241213   6.368891  -1.922   0.0546 .
Acousticness      -3.018936   1.947207  -1.550   0.1210
Instrumentalness  15.960012  17.289584   0.923   0.3560
Liveness          -2.143883   3.328173  -0.644   0.5195
Valence           -3.440900   2.475330  -1.390   0.1645
Duration           0.009314   0.010276   0.906   0.3647
Tempo              0.027366   0.019143   1.430   0.1528
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 78.597  on 67  degrees of freedom
Residual deviance: 48.603  on 55  degrees of freedom
AIC: 74.603

Number of Fisher Scoring iterations: 8
```

In [84]:

```
prev_log<-predict(fit.logb,don_test,type="response")
prev_log
```

```
20  0.959155111141887
21  0.790405663732002
22  0.980406096197397
23  0.91446117997479
24  0.758107074966233
25  0.386991486841755
26  0.940638125553914
27  0.987507137948899
28  0.886055353768732
29  0.648264980966115
30  0.830376824937906
31  0.789047748002245
32  0.999995291875837
33  0.942056324160013
34  0.801213170809787
35  0.998326323031655
36  0.965973243448102
37  0.754595467980921
38  1
39  0.967936375023105
40  0.475888096414484
41  0.597018097146328
42  0.953385273112769
43  0.95193225168022
44  0.866276120588576
45  0.172194845919893
46  0.0435694259610064
47  0.770990416843536
48  0.995800587587766
49  0.925726194264064
50  0.730027461048904
51  0.955976892713729
52  0.49170981758672
53  0.365466749806417
54  0.920154651538289
55  0.852398080708586
56  0.945360700128469
57  0.838819264648668
58  1
59  0.977575356028909
60  0.946215577053556
61  0.87239427332388
62  0.560601882604612
63  0.905773620957902
64  0.798818171878527
65  0.607289708182485
66  0.995132549492187
67  0.792292410261354
```

In [93]:

```
test_y<-don_test$Y
prev_y<-rep(0,length(test_y))
prev_y[prev_log >=0.8] =1
prev_y
test_y
```

```
1 0 1 1 0 0 1 1 1 0 1 0 1 1 1 1 1 0 1 1 0 0 1 1 1 0 0 0 1 1 0 1 0 0 1 1 1 1 1 1 1 1 0 1 0
0 1 0

0 1 1 1 0 0 1 1 0 1 1 0 1 1 0 0 1 1 0 0 0 1 1 1 1 0 0 0 1 0 1 1 1 0 1 1 1 1 0 1 0 0 1 1 1
0 1 1
```

In [94]:
```
taux_err_log = mean(prev_y != test_y)
# Matrice de confusion
confus_log<-table(prev_y,test_y)
confus_log
```

```
      test_y
prev_y  0  1
     0  9  9
     1 10 20
```

In [57]:
```
library(boot)
```

```
Warning message:
"package 'boot' was built under R version 3.6.3"
```

In [58]:
```
fit.log<-glm(Y~., family=binomial, data=don)
summary(fit.log)
```

```
Call:
glm(formula = Y ~ ., family = binomial, data = don_std)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.1402  -0.6821   0.4230   0.7021   1.8255

Coefficients:
                 Estimate Std. Error z value Pr(>|z|)
(Intercept)       0.96816    0.81050   1.195  0.23228
Danceability      0.77535    0.33692   2.301  0.02138 *
Energy           -0.80163    0.47525  -1.687  0.09165 .
Key               0.04767    0.25653   0.186  0.85257
Loudness          0.30139    0.39090   0.771  0.44070
Mode             -0.35033    0.28337  -1.236  0.21635
Speechiness      -0.70539    0.26938  -2.619  0.00883 **
Acousticness     -0.42548    0.36072  -1.180  0.23819
Instrumentalness  0.38703    0.47971   0.807  0.41978
Liveness         -0.43767    0.25931  -1.688  0.09144 .
Valence          -1.01396    0.32785  -3.093  0.00198 **
Duration         -0.67561    2.61873  -0.258  0.79641
Tempo            -0.78135    7.83008  -0.100  0.92051
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 145.25  on 115  degrees of freedom
Residual deviance: 104.81  on 103  degrees of freedom
AIC: 130.81

Number of Fisher Scoring iterations: 10
```

In [59]:
```
#Regression Logistique - 4-FOLD

set.seed(2226947)

cv.errlo1k<-cv.glm(don_std,fit.log, K=4)

cv.errlo1k$delta
```

```
Warning message:
"glm.fit: fitted probabilities numerically 0 or 1 occurred"
```

0.208909643066211 0.200356772895736

In [ ]: