```
> #Pregunta 1
> #---------------------------------------------
> #Abrimos archivo variables.csv y lo nombramos como var_df
> var_df <- read.csv("C:\\Users\\usuario\\Documents\\4. UOC\\1º Álgebra Lineal\\Reto 4\\variab
les.csv")
>
> #Imprimimos archivo para ver su contenido
> fix(var_df)
> var_df
   id  rent inc_sal inc_ret inc_emp inc_non inc_oth gini dist8020 mean_age perc_chil per_ret h
ome_size
1   1 20066   11510    2362     898     590    4707 39.7      3.4     39.5      13.8    11.6
    2.17
2   2 18811    9521    4529     502     550    3709 35.6      3.7     43.0      20.2    23.3
    2.80
3   3 20724   13116    3631     640     574    2763 31.8      2.6     41.5      17.3    17.7
    2.36
4   4 15213    9483    2663     721     595    1753 32.5      3.0     40.5      17.6    15.7
    2.35
5   5  4821    1916    1201     343    1259      44 34.2      2.8     30.7      34.1     6.4
    3.46
6   6  9553    4981    2487     613     883     589 28.5      2.8     40.0      23.4    18.5
    2.79
7   7 30340   15996    4993     574     636    8141 33.3      2.7     44.2      18.7    23.2
    2.41
8   8 25658   15572    3997     656     796    4636 35.1      3.2     41.8      17.4    17.0
    2.30
9   9 25572   12486    5688     562     658    6177 33.4      2.7     44.6      17.4    24.4
    2.40
10 10 24310   12519    5053     613     618    5508 33.7      3.1     44.7      15.9    24.2
    2.35
11 11 26194   14825    5165     643     772    4788 30.0      2.7     43.4      17.2    23.0
    2.26
12 12 22029   11395    5342     564     501    4228 34.0      3.1     43.9      17.0    22.9
    2.38
13 13 25916   14898    4505     590     660    5264 35.5      3.0     41.8      18.0    18.0
    2.56
14 14 25471   13482    4829     641     607    5912 32.9      2.6     43.2      16.6    19.9
    2.47
15 15 19826   12336    4051     707     615    2117 29.3      2.6     42.8      13.5    16.3
    2.50
16 16 19866   13906    3214     677     523    1546 27.2      2.3     40.5      18.7    14.3
    2.40
17 17 17174   12461    2285     763     516    1150 30.4      2.7     37.5      20.7    10.9
    2.51
18 18 24186   12350    6541     549     647    4100 32.1      2.9     46.9      15.1    29.4
    2.38
19 19 22362   10573    5842     665     524    4758 37.0      3.4     43.9      17.4    24.7
    2.33
20 20 15093    9193    2694     738     741    1728 31.4      2.9     40.4      19.1    15.7
    2.67
21 21 28792   18156    4500     606     510    5020 30.3      2.4     41.2      19.0    16.5
    2.78
22 22 34639   17957    8228     375     441    7638 30.5      2.6     45.0      15.4    22.8
    3.01
23 23 24488   15370    3835     689     493    4101 32.1      2.6     41.6      19.9    18.4
    3.04
24 24 22618   15689    3305     702     452    2470 26.4      2.3     40.4      21.6    15.2
    2.56
25 25 22451   16204    2405     648     515    2679 27.3      2.3     37.5      24.1    10.4
    2.66
26 26 31563   16706    4633     548     592    9083 38.8      3.1     41.5      18.0    17.9
    2.84
27 27 20447   12889    4072     615     625    2247 30.9      2.7     41.7      18.8    18.4
    2.49
28 28 15831   12302    1110     685     643    1090 29.6      2.5     33.3      26.5     6.1
    2.63
29 29 22824   11759    4407     733     572    5352 36.9      3.5     41.7      15.8    18.6
    2.22
30 30 26607   13465    5283     731     598    6530 33.9      2.8     43.7      13.9    21.0
    2.15
31 31 22107   13497    3227     660     690    4032 33.4      3.2     40.3      16.8    14.7
    2.36
32 32 25540   15790    4022     539     584    4606 27.5      2.4     42.1      17.7    17.1
```

```
    2.51
33 33 26814   16411    4800     520     579    4503 27.7      2.4     41.2       19.8    16.1
    2.53
34 34 18480   10741    4747     550     741    1700 32.2      3.0     42.8       19.0    21.0
    2.62
35 35 12225    7500    2886     625     528     686 33.1      2.9     40.6       18.0    16.9
    2.69
36 36 13462    9315    2204     728     517     697 30.4      2.7     39.0       18.2    12.7
    2.73
37 37  9395    5533    2333     627     568     335 30.5      2.7     38.8       22.1    16.0
    3.13
38 38 13732    9090    2441     701     653     846 31.1      2.7     38.5       20.4    13.4
    2.95
39 39 11387    7136    2352     715     537     647 31.8      2.6     38.1       21.7    14.1
    2.96
40 40 11536    7047    2601     722     524     642 31.5      3.0     38.9       20.8    14.4
    3.11
41 41 10450    5730    2963     633     682     443 28.9      2.7     40.2       20.8    18.6
    3.04
42 42 10219    6626    1970     625     514     484 32.4      3.0     37.1       21.5    12.0
    3.17
43 43 13729    7526    3984     663     550    1007 32.2      3.0     42.0       17.2    21.0
    2.57
44 44 15300    8720    4254     691     549    1086 31.4      2.7     42.8       16.7    20.5
    2.71
45 45 13898    9605    2169     693     567     865 29.0      2.6     39.7       19.6    12.3
    2.69
46 46 12539    8973    1672     759     514     621 29.8      2.9     37.1       22.1    10.0
    2.83
47 47 22855   15290    3271     665     630    2999 29.2      2.4     39.5       19.2    12.8
    2.49
48 48 22607   14163    4685     675     672    2411 30.6      2.7     42.3       16.0    18.5
    2.56
49 49 19766   11983    2052     937     440    4354 37.0      3.4     39.3       14.0    10.4
    2.10
50 50 18049   10907    2633     765     644    3099 39.3      3.4     40.7       17.3    14.2
    2.26
51 51 31114   16551    6118     490     596    7359 31.9      2.5     44.2       17.6    22.6
    2.85
52 52 14229    9066    2829     732     664     937 31.6      2.8     41.9       16.6    17.6
    2.32
53 53 13592    8303    2734     751     613    1190 28.0      2.7     39.9       20.1    14.8
    2.63
54 54 19252   11109    3942     680     563    2957 28.3      2.3     42.7       20.7    20.1
    2.57
55 55 12003    6413    2883     719     804    1185 33.7      3.1     41.4       21.0    19.5
    2.85
56 56 23778   17076    2901     756     499    2547 25.4      2.1     39.8       20.5    11.4
    2.73
57 57 17836   11784    3660     633     657    1102 24.9      2.1     42.0       18.6    15.8
    2.69
58 58 15530   10457    2544     750     736    1044 25.4      2.2     39.8       22.0    15.5
    2.70
59 59 15116    9763    3083     694     590     985 24.1      2.1     41.4       18.4    16.0
    2.75
60 60 16783   10478    3350     663     708    1584 28.1      2.5     41.6       19.0    16.2
    2.75
61 61 13650    7382    3994     709     835     730 26.5      2.3     45.9       12.7    26.7
    2.13
>
> #Comprobamos que var_df es de tipo dataframe
> class(var_df)
[1] "data.frame"
>
> #Eliminamos la primera columna llamada 'id'. Para ello vamos a usar el paquete "dplyr"
> library(dplyr)
> var_df <- select(var_df, -id)
>
> #Comprobamos que realmente se ha eliminado la columna
> fix(var_df)
> var_df
    rent inc_sal inc_ret inc_emp inc_non inc_oth gini dist8020 mean_age perc_chil per_ret home
_size
1  20066   11510    2362     898     590    4707 39.7      3.4     39.5       13.8    11.6
```

```
 2.17
2  18811    9521    4529     502     550    3709 35.6       3.7      43.0       20.2    23.3
 2.80
3  20724   13116    3631     640     574    2763 31.8       2.6      41.5       17.3    17.7
 2.36
4  15213    9483    2663     721     595    1753 32.5       3.0      40.5       17.6    15.7
 2.35
5   4821    1916    1201     343    1259      44 34.2       2.8      30.7       34.1     6.4
 3.46
6   9553    4981    2487     613     883     589 28.5       2.8      40.0       23.4    18.5
 2.79
7  30340   15996    4993     574     636    8141 33.3       2.7      44.2       18.7    23.2
 2.41
8  25658   15572    3997     656     796    4636 35.1       3.2      41.8       17.4    17.0
 2.30
9  25572   12486    5688     562     658    6177 33.4       2.7      44.6       17.4    24.4
 2.40
10 24310   12519    5053     613     618    5508 33.7       3.1      44.7       15.9    24.2
 2.35
11 26194   14825    5165     643     772    4788 30.0       2.7      43.4       17.2    23.0
 2.26
12 22029   11395    5342     564     501    4228 34.0       3.1      43.9       17.0    22.9
 2.38
13 25916   14898    4505     590     660    5264 35.5       3.0      41.8       18.0    18.0
 2.56
14 25471   13482    4829     641     607    5912 32.9       2.6      43.2       16.6    19.9
 2.47
15 19826   12336    4051     707     615    2117 29.3       2.6      42.8       13.5    16.3
 2.50
16 19866   13906    3214     677     523    1546 27.2       2.3      40.5       18.7    14.3
 2.40
17 17174   12461    2285     763     516    1150 30.4       2.7      37.5       20.7    10.9
 2.51
18 24186   12350    6541     549     647    4100 32.1       2.9      46.9       15.1    29.4
 2.38
19 22362   10573    5842     665     524    4758 37.0       3.4      43.9       17.4    24.7
 2.33
20 15093    9193    2694     738     741    1728 31.4       2.9      40.4       19.1    15.7
 2.67
21 28792   18156    4500     606     510    5020 30.3       2.4      41.2       19.0    16.5
 2.78
22 34639   17957    8228     375     441    7638 30.5       2.6      45.0       15.4    22.8
 3.01
23 24488   15370    3835     689     493    4101 32.1       2.6      41.6       19.9    18.4
 3.04
24 22618   15689    3305     702     452    2470 26.4       2.3      40.4       21.6    15.2
 2.56
25 22451   16204    2405     648     515    2679 27.3       2.3      37.5       24.1    10.4
 2.66
26 31563   16706    4633     548     592    9083 38.8       3.1      41.5       18.0    17.9
 2.84
27 20447   12889    4072     615     625    2247 30.9       2.7      41.7       18.8    18.4
 2.49
28 15831   12302    1110     685     643    1090 29.6       2.5      33.3       26.5     6.1
 2.63
29 22824   11759    4407     733     572    5352 36.9       3.5      41.7       15.8    18.6
 2.22
30 26607   13465    5283     731     598    6530 33.9       2.8      43.7       13.9    21.0
 2.15
31 22107   13497    3227     660     690    4032 33.4       3.2      40.3       16.8    14.7
 2.36
32 25540   15790    4022     539     584    4606 27.5       2.4      42.1       17.7    17.1
 2.51
33 26814   16411    4800     520     579    4503 27.7       2.4      41.2       19.8    16.1
 2.53
34 18480   10741    4747     550     741    1700 32.2       3.0      42.8       19.0    21.0
 2.62
35 12225    7500    2886     625     528     686 33.1       2.9      40.6       18.0    16.9
 2.69
36 13462    9315    2204     728     517     697 30.4       2.7      39.0       18.2    12.7
 2.73
37  9395    5533    2333     627     568     335 30.5       2.7      38.8       22.1    16.0
 3.13
38 13732    9090    2441     701     653     846 31.1       2.7      38.5       20.4    13.4
```

```
 2.95
39 11387    7136    2352     715     537     647 31.8      2.6      38.1       21.7    14.1
 2.96
40 11536    7047    2601     722     524     642 31.5      3.0      38.9       20.8    14.4
 3.11
41 10450    5730    2963     633     682     443 28.9      2.7      40.2       20.8    18.6
 3.04
42 10219    6626    1970     625     514     484 32.4      3.0      37.1       21.5    12.0
 3.17
43 13729    7526    3984     663     550    1007 32.2      3.0      42.0       17.2    21.0
 2.57
44 15300    8720    4254     691     549    1086 31.4      2.7      42.8       16.7    20.5
 2.71
45 13898    9605    2169     693     567     865 29.0      2.6      39.7       19.6    12.3
 2.69
46 12539    8973    1672     759     514     621 29.8      2.9      37.1       22.1    10.0
 2.83
47 22855   15290    3271     665     630    2999 29.2      2.4      39.5       19.2    12.8
 2.49
48 22607   14163    4685     675     672    2411 30.6      2.7      42.3       16.0    18.5
 2.56
49 19766   11983    2052     937     440    4354 37.0      3.4      39.3       14.0    10.4
 2.10
50 18049   10907    2633     765     644    3099 39.3      3.4      40.7       17.3    14.2
 2.26
51 31114   16551    6118     490     596    7359 31.9      2.5      44.2       17.6    22.6
 2.85
52 14229    9066    2829     732     664     937 31.6      2.8      41.9       16.6    17.6
 2.32
53 13592    8303    2734     751     613    1190 28.0      2.7      39.9       20.1    14.8
 2.63
54 19252   11109    3942     680     563    2957 28.3      2.3      42.7       20.7    20.1
 2.57
55 12003    6413    2883     719     804    1185 33.7      3.1      41.4       21.0    19.5
 2.85
56 23778   17076    2901     756     499    2547 25.4      2.1      39.8       20.5    11.4
 2.73
57 17836   11784    3660     633     657    1102 24.9      2.1      42.0       18.6    15.8
 2.69
58 15530   10457    2544     750     736    1044 25.4      2.2      39.8       22.0    15.5
 2.70
59 15116    9763    3083     694     590     985 24.1      2.1      41.4       18.4    16.0
 2.75
60 16783   10478    3350     663     708    1584 28.1      2.5      41.6       19.0    16.2
 2.75
61 13650    7382    3994     709     835     730 26.5      2.3      45.9       12.7    26.7
 2.13
>
> #Convertimos el vector var_df en una matriz
> var_matrix <- as.matrix(var_df)
>
> #Comprobamos que ahora el vector var_df es una matriz
> class(var_matrix)
[1] "matrix" "array"
>
> #Buscamos la dimensión que tiene la matriz, y podemos observar que nos devuelve
> # 61, 12, es decir, contamos con 61 observaciones que coinciden con el número de
> #secciones censales, y 12 columnas que coinciden con el número de variables:
> dim(var_matrix)
[1] 61 12
>
> #-----------------------------------------------------
> #Pregunta 2
> #-----------------------------------------------------
>
> #En este intento, la variable V es igual a la variable 'mean_age'. Para calcular
> #la razón de 'mean_age' vamos a detectar los valores máximo y mínimo de la variable.
>
> #Asignamos al vector V la columna 'mean_age' de la matriz var_df
> V <- var_matrix[,9]
> #Definimos vector máximo como el valor máximo de V, y definimos el vector mínimo
> #como el valor mínimo de V
> maximo <- max(V)
> minimo <- min(V)
```

```
> fix(maximo)
> maximo
[1] 46.9
> fix(minimo)
> minimo
[1] 30.7
>
> #Calculamos la razón entre el valor Máximo y el valor Mínimo de V (M/m)
> razon <- maximo/minimo
> fix(razon)
> razon
[1] 1.527687
> #Redondeamos resultado a dos decimales
> round(razon, digits=2)
[1] 1.53
>
> #------------------------------------
> #Pregunta 3
> #-------------------------------------
>
> #Calculamos la matriz de datos normalizada y la guardamos en la variable Xs
> Xs <- as.matrix(scale(var_matrix, center = TRUE, scale = TRUE))
> fix(Xs)
> Xs
          rent     inc_sal      inc_ret      inc_emp      inc_non     inc_oth        gini   di
st8020   mean_age    perc_chil      per_ret   home_size
1   0.11215097 -0.00395886 -0.931139719  2.418143014 -0.214973914  0.79806710  2.39053026  1.7
676480 -0.5671366 -1.512406683 -1.169959532 -1.54065428
2  -0.08473303 -0.55542215  0.645412259 -1.550917436 -0.530200087  0.35484048  1.21938116  2.5
907512  0.7141721  0.414628464  1.302649642  0.66528253
3   0.21537779  0.44131517 -0.007907388 -0.167760007 -0.341064383 -0.06529216  0.13392589 -0.4
272940  0.1650398 -0.458559337  0.119178584 -0.87537175
4  -0.64918611 -0.56595790 -0.712153958  0.644093267 -0.175570643 -0.51384816  0.33387817  0.6
701770 -0.2010484 -0.368229565 -0.303489651 -0.91038662
5  -2.27947972 -2.66395824 -1.775799086 -3.144555344  5.057183822 -1.27284043  0.81947658  0.1
214415 -3.7887128  4.599907925 -2.268896942  2.97626394
6  -1.53712508 -1.81416690 -0.840198788 -0.438377765  2.094057800 -1.03079784 -0.80870632  0.1
214415 -0.3840925  1.378146038  0.288245878  0.63026766
7   1.72393277  1.23981404  0.982984995 -0.829270082  0.147536184  2.32315748  0.56239507 -0.1
529263  1.1534779 -0.037020399  1.281516230 -0.70029740
8   0.98942212  1.12225726  0.258367657 -0.007393928  1.408440874  0.76653494  1.07655810  1.2
189125  0.2748662 -0.428449413 -0.028755298 -1.08546097
9   0.97593047  0.26664354  1.488616571 -0.949544641  0.320910579  1.45091592  0.59095969 -0.1
529263  1.2999132 -0.428449413  1.535117171 -0.73531227
10  0.77794832  0.27579300  1.026636642 -0.438377765  0.005684406  1.15380309  0.67665352  0.9
445447  1.3365220 -0.880098276  1.492850347 -0.91038662
11  1.07350963  0.91514662  1.108119716 -0.137691367  1.219305171  0.83404040 -0.38023714 -0.1
529263  0.8606074 -0.488669261  1.239249406 -1.22552045
12  0.42010577 -0.03584336  1.236892074 -0.929498881 -0.916352148  0.58533609  0.76234736  0.9
445447  1.0436515 -0.548889110  1.218115995 -0.80534201
13  1.02989708  0.93538635  0.627951601 -0.668904003  0.336671888  1.04543907  1.19081654  0.6
701770  0.2748662 -0.247789868  0.182578819 -0.17507435
14  0.96008562  0.54279107  0.863670494 -0.157737127 -0.081002791  1.33322549  0.44813662 -0.4
272940  0.7873897 -0.669328806  0.584113643 -0.49020818
15  0.07449984  0.22505505  0.297654139  0.503772948 -0.017957557 -0.35219035 -0.58018942 -0.4
272940  0.6409544 -1.602736456 -0.176689180 -0.38516357
16  0.08077503  0.66034784 -0.311286334  0.203086551 -0.742977753 -0.60577993 -1.18004628 -1.2
503973 -0.2010484 -0.037020399 -0.599357415 -0.73531227
17 -0.34154506  0.25971212 -0.987159333  1.065054224 -0.798142334 -0.78164941 -0.26597869 -0.1
529263 -1.2993131  0.565178085 -1.317893414 -0.35014870
18  0.75849524  0.22893665  2.109197484 -1.079842080  0.234223381  0.52848939  0.21961972  0.3
958092  2.1419161 -1.120977669  2.591787758 -0.80534201
19  0.47234671 -0.26374825  1.600655798  0.082811992 -0.735097099  0.82071696  1.61928573  1.7
676480  1.0436515 -0.428449413  1.598517406 -0.98041636
20 -0.66801167 -0.64636230 -0.689600607  0.814482226  0.975004887 -0.52495103  0.01966744  0.3
958092 -0.2376573  0.083419298 -0.303489651  0.21008922
21  1.48108303  1.83868820  0.624313963 -0.508537924 -0.845426260  0.93707505 -0.29454330 -0.9
760295  0.0552133  0.053309374 -0.134422357  0.59525279
22  2.39835851  1.78351415  3.336536288 -2.823823186 -1.389191407  2.09976771 -0.23741407 -0.4
272940  1.4463485 -1.030647896  1.196982583  1.40059480
23  0.80587290  1.06625144  0.140508211  0.323361110 -0.979397383  0.52893350  0.21961972 -0.4
272940  0.2016486  0.324298692  0.267112466  1.50563941
24  0.51250790  1.15469628 -0.245081336  0.453658549 -1.302504210 -0.19541781 -1.40856318 -1.2
503973 -0.2376573  0.836167403 -0.409156709 -0.17507435
```

```
25  0.48630900  1.29748341 -0.899856039 -0.087576967 -0.806022988 -0.10259781 -1.15148167 -1.2
503973 -1.2993131  1.588915507 -1.423560473  0.17507435
26  1.91579662  1.43666620  0.721075114 -1.089864960 -0.199212606  2.74151367  2.13344875  0.9
445447  0.1650398 -0.247789868  0.161445408  0.80534201
27  0.17192212  0.37837793  0.312932216 -0.418332005  0.060848987 -0.29445542 -0.12315562 -0.1
529263  0.2382574 -0.006910474  0.267112466 -0.42017844
28 -0.55223447  0.21562833 -1.842004084  0.283269590  0.202700764 -0.80829630 -0.49449558 -0.7
016618 -2.8368835  2.311553687 -2.332297178  0.07002974
29  0.54482512  0.06507802  0.556653911  0.764367826 -0.356825692  1.08452117  1.59072112  2.0
420157  0.2382574 -0.910208200  0.309379290 -1.36557993
30  1.13830094  0.53807770  1.193967955  0.744322066 -0.151928680  1.60768846  0.73378275  0.1
214415  0.9704338 -1.482296759  0.816581172 -1.61068401
31  0.43234239  0.54694991 -0.301828477  0.032697592  0.573091517  0.49828958  0.59095969  1.2
189125 -0.2742661 -0.609108958 -0.514823768 -0.87537175
32  0.97091032  1.18269919  0.276555843 -1.180070879 -0.262257840  0.75321150 -1.09435244 -0.9
760295  0.3846927 -0.338119640 -0.007621886 -0.35014870
33  1.17077503  1.35487551  0.842572198 -1.370505597 -0.301661112  0.70746767 -1.03722322 -0.9
760295  0.0552133  0.294188767 -0.218956004 -0.28011896
34 -0.13666020 -0.21716915  0.804013243 -1.069819200  0.975004887 -0.53738624  0.24818434  0.6
701770  0.6409544  0.053309374  0.816581172  0.03501487
35 -1.11794258 -1.11575764 -0.549915337 -0.318103205 -0.703574482 -0.98771870  0.50526585  0.3
958092 -0.1644396 -0.247789868 -0.049888710  0.28011896
36 -0.92388242 -0.61253700 -1.046089056  0.714253427 -0.790261679 -0.98283344 -0.26597869 -0.1
529263 -0.7501807 -0.187570019 -0.937492003  0.42017844
37 -1.56191207 -1.66112129 -0.952238015 -0.298057446 -0.388348309 -1.14360301 -0.23741407 -0.1
529263 -0.8233984  0.986717024 -0.240089415  1.82077323
38 -0.88152491 -0.67491972 -0.873665051  0.443635669  0.281507307 -0.91666032 -0.06602640 -0.1
529263 -0.9332248  0.474848312 -0.789558121  1.19050558
39 -1.24940775 -1.21667903 -0.938414994  0.583955988 -0.632648593 -1.00503918  0.13392589 -0.4
272940 -1.0796601  0.866277327 -0.641624238  1.22552045
40 -1.22603268 -1.24135486 -0.757260659  0.654116147 -0.735097099 -1.00725975  0.04823205  0.6
701770 -0.7867896  0.595288009 -0.578224003  1.75074349
41 -1.39640401 -1.60650174 -0.493895723 -0.237920166  0.510046282 -1.09563861 -0.69444787 -0.1
529263 -0.3108749  0.595288009  0.309379290  1.50563941
42 -1.43264322 -1.35807987 -1.216330479 -0.318103205 -0.813903642 -1.07742990  0.30531356  0.6
701770 -1.4457483  0.806057478 -1.085425885  1.96083271
43 -0.88199555 -1.10854897  0.248909800  0.062766232 -0.530200087 -0.84515783  0.24818434  0.6
701770  0.3480839 -0.488669261  0.816581172 -0.14005948
44 -0.63553758 -0.77750465  0.445342211  0.343406870 -0.538080741 -0.81007276  0.01966744 -0.1
529263  0.6409544 -0.639218882  0.710914113  0.35014870
45 -0.85548288 -0.53213260 -1.071552517  0.363452629 -0.396228964 -0.90822214 -0.66588326 -0.4
272940 -0.4939190  0.233968919 -1.022025650  0.28011896
46 -1.06868237 -0.70735874 -1.433133658  1.024962704 -0.813903642 -1.01658616 -0.43736636  0.3
958092 -1.4457483  0.986717024 -1.508094120  0.77032714
47  0.54968839  1.04407092 -0.269817270  0.082811992  0.100252258  0.03951894 -0.60875403 -0.9
760295 -0.5671366  0.113529222 -0.916358591 -0.42017844
48  0.51078223  0.73160278  0.758906541  0.183040791  0.431239739 -0.22162059 -0.20884946 -0.1
529263  0.4579103 -0.849988351  0.288245878 -0.17507435
49  0.06508706  0.12718349 -1.156673228  2.809035331 -1.397072061  0.64129456  1.61928573  1.7
676480 -0.6403543 -1.452186835 -1.423560473 -1.78575836
50 -0.20427534 -0.17114456 -0.733979781  1.085099984  0.210581419  0.08393043  2.27627181  1.7
676480 -0.1278308 -0.458559337 -0.620490827 -1.22552045
51  1.84535764  1.39369143  1.801453373 -1.671191995 -0.167689989  1.97585967  0.16249050 -0.7
016618  1.1534779 -0.368229565  1.154715760  0.84035688
52 -0.80355571 -0.68157388 -0.591384401  0.754344946  0.368194505 -0.87624587  0.07679666  0.1
214415  0.3114750 -0.669328806  0.098045173 -1.01543123
53 -0.90348806 -0.89312063 -0.660499509  0.944779665 -0.033718865 -0.76388482 -0.95152938 -0.1
529263 -0.4207014  0.384518540 -0.493690356  0.07002974
54 -0.01554909 -0.11513874  0.218353648  0.233155190 -0.427751581  0.02086612 -0.86583555 -1.2
503973  0.6043456  0.565178085  0.626380466 -0.14005948
55 -1.15276987 -1.41713552 -0.552097919  0.624047507  1.471486109 -0.76610539  0.67665352  0.9
445447  0.1284309  0.655507858  0.499579996  0.84035688
56  0.69448833  1.53925112 -0.539002425  0.994894065 -0.932113457 -0.16122097 -1.69420930 -1.7
991328 -0.4573102  0.504958237 -1.212226355  0.42017844
57 -0.23769071  0.07200944  0.013190907 -0.237920166  0.313029925 -0.80296692 -1.83703237 -1.7
991328  0.3480839 -0.067130323 -0.282356239  0.28011896
58 -0.59945525 -0.29591001 -0.798729724  0.934756785  0.935601615 -0.82872558 -1.69420930 -1.5
247650 -0.4573102  0.956607099 -0.345756474  0.31513383
59 -0.66440344 -0.48832606 -0.406592430  0.373475509 -0.214973914 -0.85492836 -2.06554926 -1.7
991328  0.1284309 -0.127350171 -0.240089415  0.49020818
60 -0.40288501 -0.29008762 -0.212342601  0.062766232  0.714943295 -0.58890357 -0.92296477 -0.7
016618  0.2016486  0.053309374 -0.197822592  0.49020818
61 -0.89438904 -1.14847392  0.256185075  0.523818708  1.715786392 -0.96817765 -1.37999857 -1.2
503973  1.7758279 -1.843615849  2.021185641 -1.68071375
```

```
>
> #Comprobamos que realmente el valor de la media de los datos de Xs es
> #prácticamente cero y el valor de la desviación típica prácticamente 1.
> mean(Xs)
[1] 4.598972e-19
> sd(Xs)
[1] 0.9924475
> round(mean(Xs), digits = 0)
[1] 0
> round(sd(Xs), digits = 0)
[1] 1
>
> #Ahora vamos a calcular la matriz de covarianza de los datos Xs y la vamos a
> #guardar en la variable CXs.
> CXs <- cov(Xs)
> fix(CXs)
> CXs
                 rent     inc_sal     inc_ret     inc_emp     inc_non    inc_oth        gini
  dist8020    mean_age  perc_chil     per_ret  home_size
rent       1.00000000  0.92673026  0.76122489 -0.28365656 -0.31599008  0.9130952  0.18017222 -
0.04145366  0.57540695 -0.4716990  0.41627073 -0.3950107
inc_sal    0.92673026  1.00000000  0.52982883 -0.13254741 -0.39950272  0.7277385 -0.03060094 -
0.23533662  0.37060666 -0.3548565  0.14320960 -0.3646303
inc_ret    0.76122489  0.52982883  1.00000000 -0.51158121 -0.16092694  0.7283159  0.16273694
0.05805812  0.84615522 -0.5501134  0.83398303 -0.2723374
inc_emp   -0.28365656 -0.13254741 -0.51158121  1.00000000 -0.29219282 -0.3048800  0.01873346
0.09633633 -0.16870717 -0.2774858 -0.35580857 -0.3969711
inc_non   -0.31599008 -0.39950272 -0.16092694 -0.29219282  1.00000000 -0.2019849  0.02066379
0.02290216 -0.19666356  0.3776646 -0.00395581  0.1294757
inc_oth    0.91309520  0.72773849  0.72831587 -0.30487998 -0.20198487  1.0000000  0.45736948
0.21856466  0.53895303 -0.4420784  0.45691555 -0.3588892
gini       0.18017222 -0.03060094  0.16273694  0.01873346  0.02066379  0.4573695  1.00000000
0.87938833  0.07983447 -0.2432927  0.14512320 -0.2259754
dist8020  -0.04145366 -0.23533662  0.05805812  0.09633633  0.02290216  0.2185647  0.87938833
1.00000000  0.05947531 -0.2204091  0.14831518 -0.2113529
mean_age   0.57540695  0.37060666  0.84615522 -0.16870717 -0.19666356  0.5389530  0.07983447
0.05947531  1.00000000 -0.7753111  0.91512413 -0.4825750
perc_chil -0.47169903 -0.35485651 -0.55011339 -0.27748580  0.37766463 -0.4420784 -0.24329266 -
0.22040906 -0.77531110  1.0000000 -0.55090986  0.6871720
per_ret    0.41627073  0.14320960  0.83398303 -0.35580857 -0.00395581  0.4569156  0.14512320
0.14831518  0.91512413 -0.5509099  1.00000000 -0.3038743
home_size -0.39501073 -0.36463033 -0.27233745 -0.39697106  0.12947569 -0.3588892 -0.22597540 -
0.21135293 -0.48257496  0.6871720 -0.30387429  1.0000000
>
> #Visualizamos la matriz como una imagen y la guardamos en formato .jpeg
> image(CXs)
> jpeg('CXs.jpeg')
> image(CXs)
> dev.off()
windows
      2
>
> #Buscamos el nombre (índice) de las variables que contienen el mayor y menor
> #valor absoluto de covarianza.
> #Primero mostramos la matriz CXs en valores absolutos
>
> CXs_ab <- abs(CXs)
> CXs_ab
                rent    inc_sal    inc_ret    inc_emp    inc_non    inc_oth       gini   dist80
20   mean_age perc_chil    per_ret home_size
rent      1.00000000 0.92673026 0.76122489 0.28365656 0.31599008 0.9130952 0.18017222 0.041453
66 0.57540695 0.4716990 0.41627073 0.3950107
inc_sal   0.92673026 1.00000000 0.52982883 0.13254741 0.39950272 0.7277385 0.03060094 0.235336
62 0.37060666 0.3548565 0.14320960 0.3646303
inc_ret   0.76122489 0.52982883 1.00000000 0.51158121 0.16092694 0.7283159 0.16273694 0.058058
12 0.84615522 0.5501134 0.83398303 0.2723374
inc_emp   0.28365656 0.13254741 0.51158121 1.00000000 0.29219282 0.3048800 0.01873346 0.096336
33 0.16870717 0.2774858 0.35580857 0.3969711
inc_non   0.31599008 0.39950272 0.16092694 0.29219282 1.00000000 0.2019849 0.02066379 0.022902
16 0.19666356 0.3776646 0.00395581 0.1294757
inc_oth   0.91309520 0.72773849 0.72831587 0.30487998 0.20198487 1.0000000 0.45736948 0.218564
66 0.53895303 0.4420784 0.45691555 0.3588892
gini      0.18017222 0.03060094 0.16273694 0.01873346 0.02066379 0.4573695 1.00000000 0.879388
33 0.07983447 0.2432927 0.14512320 0.2259754
```

```
dist8020  0.04145366 0.23533662 0.05805812 0.09633633 0.02290216 0.2185647 0.87938833 1.000000
00 0.05947531 0.2204091 0.14831518 0.2113529
mean_age  0.57540695 0.37060666 0.84615522 0.16870717 0.19666356 0.5389530 0.07983447 0.059475
31 1.00000000 0.7753111 0.91512413 0.4825750
perc_chil 0.47169903 0.35485651 0.55011339 0.27748580 0.37766463 0.4420784 0.24329266 0.220409
06 0.77531110 1.0000000 0.55090986 0.6871720
per_ret   0.41627073 0.14320960 0.83398303 0.35580857 0.00395581 0.4569156 0.14512320 0.148315
18 0.91512413 0.5509099 1.00000000 0.3038743
home_size 0.39501073 0.36463033 0.27233745 0.39697106 0.12947569 0.3588892 0.22597540 0.211352
93 0.48257496 0.6871720 0.30387429 1.0000000
> #Mostramos el par de valores máximo y mínimo de los valores absolutos de la
> #matriz CXs_ab.
> par_max = tail(sort(CXs_ab), 4)
> par_min = head(sort(CXs_ab), 4)
> par_max
[1] 1 1 1 1
> par_min
[1] 0.00395581 0.00395581 0.01873346 0.01873346
>
> #Convertimos la matriz CXs_ab en una lista ordenada y localizamos el par de
> #valores mínimos y el par de valores máximos
> lista_ord = as.list(CXs_ab)
> pos_par_max = which(lista_ord == par_max)
> pos_par_min_1 = which(lista_ord == par_min[1])
> pos_par_min_2 = which(lista_ord == par_min[3])
> pos_par_max
[1]  53  92 105 118
> pos_par_min_1
[1]  59 125
> pos_par_min_2
[1] 43 76
> #Comprobamos visualmente en la lista que los datos obtenidos concuerdan
> lista_ord[53]
[[1]]
[1] 1

> lista_ord[92]
[[1]]
[1] 1

> lista_ord[59]
[[1]]
[1] 0.00395581

> lista_ord[76]
[[1]]
[1] 0.01873346

>
>
>
>
```