Practica1-intento2

2023-06-01

- Se instala la librería install.packages("reader") y cargamos la librería readr
- Creamos var_df desde la ubicación de nuestro fichero invirtiendo las barras /

```
library(reader)
```

```
## Loading required package: NCmisc
```

```
##
## Attaching package: 'reader'
```

```
## The following objects are masked from 'package:NCmisc':
##
##     cat.path, get.ext, rmv.ext
```

- Leemos el fichero

```
var_df <- read.csv('C:/Users/Manuel/Desktop/UOC/SEMESTRE 2/Algebra lineal/PRACTICA1/variables.csv')
```

- Comprobamos que el dataframe var_df carga correctamente e imprimos solo los 10 primeros valores

```
df
```

```
## function (x, df1, df2, ncp, log = FALSE)
## {
##     if (missing(ncp))
##         .Call(C_df, x, df1, df2, log)
##     else .Call(C_dnf, x, df1, df2, ncp, log)
## }
## <bytecode: 0x000002ba28fd41d8>
## <environment: namespace:stats>
```

```
head(var_df, 10)
```

```
##    id  rent inc_sal inc_ret inc_emp inc_non inc_oth gini dist8020 mean_age
## 1   1 20066   11510    2362     898     590    4707 39.7      3.4     39.5
## 2   2 18811    9521    4529     502     550    3709 35.6      3.7     43.0
## 3   3 20724   13116    3631     640     574    2763 31.8      2.6     41.5
## 4   4 15213    9483    2663     721     595    1753 32.5      3.0     40.5
## 5   5  4821    1916    1201     343    1259      44 34.2      2.8     30.7
## 6   6  9553    4981    2487     613     883     589 28.5      2.8     40.0
```

```
## 7   7 30340   15996    4993     574     636    8141 33.3      2.7     44.2
## 8   8 25658   15572    3997     656     796    4636 35.1      3.2     41.8
## 9   9 25572   12486    5688     562     658    6177 33.4      2.7     44.6
## 10 10 24310   12519    5053     613     618    5508 33.7      3.1     44.7
##    perc_chil per_ret home_size
## 1       13.8    11.6      2.17
## 2       20.2    23.3      2.80
## 3       17.3    17.7      2.36
## 4       17.6    15.7      2.35
## 5       34.1     6.4      3.46
## 6       23.4    18.5      2.79
## 7       18.7    23.2      2.41
## 8       17.4    17.0      2.30
## 9       17.4    24.4      2.40
## 10      15.9    24.2      2.35
```

- Convertimos var_df a una matriz x eliminando la primera columna. Se usa la siguiente referencia - https://stat.ethz.ch/pipermail/r-help/2006-March/102549.html
- Se imprime los primeros 10 valores para comprobar que ID ya no aparece y la clase.

```
x <- as.matrix(var_df[, -1])
head(x, 10)
```

```
##        rent inc_sal inc_ret inc_emp inc_non inc_oth gini dist8020 mean_age
##  [1,] 20066   11510    2362     898     590    4707 39.7      3.4     39.5
##  [2,] 18811    9521    4529     502     550    3709 35.6      3.7     43.0
##  [3,] 20724   13116    3631     640     574    2763 31.8      2.6     41.5
##  [4,] 15213    9483    2663     721     595    1753 32.5      3.0     40.5
##  [5,]  4821    1916    1201     343    1259      44 34.2      2.8     30.7
##  [6,]  9553    4981    2487     613     883     589 28.5      2.8     40.0
##  [7,] 30340   15996    4993     574     636    8141 33.3      2.7     44.2
##  [8,] 25658   15572    3997     656     796    4636 35.1      3.2     41.8
##  [9,] 25572   12486    5688     562     658    6177 33.4      2.7     44.6
## [10,] 24310   12519    5053     613     618    5508 33.7      3.1     44.7
##       perc_chil per_ret home_size
##  [1,]      13.8    11.6      2.17
##  [2,]      20.2    23.3      2.80
##  [3,]      17.3    17.7      2.36
##  [4,]      17.6    15.7      2.35
##  [5,]      34.1     6.4      3.46
##  [6,]      23.4    18.5      2.79
##  [7,]      18.7    23.2      2.41
##  [8,]      17.4    17.0      2.30
##  [9,]      17.4    24.4      2.40
## [10,]      15.9    24.2      2.35
```

```
class (x)
```

```
## [1] "matrix" "array"
```

- **EJERCICIO - 1**
- ***Referencias*** **https://stackoverflow.com/questions/2470248/write-lines-of-text-to-a-file-in-r**

-El ejercicio nos indica ¿Cuántas secciones censales tiene la ciudad?, como nos indica la práctica *"Por eso, se os proporcionan las siguientes variables agregadas para cada sección censal (división de la localidad por lugar de votación)."* Se refiera al número de filas que contiene la matriz, que son **61**.

```
filas <- nrow(x)
filas
```

```
## [1] 61
```

```
writeLines(paste("El número de censales que tiene la ciudad es de:", filas))
```

```
## El número de censales que tiene la ciudad es de: 61
```

- **EJERCICIO - 2**
- El ejercicio 2 nos pide ¿Cuál es la relación entre el máximo y el mínimo de V? Redondear el resultado a dos decimales.
- V = mean_age según el ejercicio. Creamos la variable con el contenido de x de edad media, buscamos el máximo y mínimo. Creamos una variable con la división y tenemos el resultado redondeado de **1.65**.

```
mean_age <- x[, "gini"]
maximo <- max(mean_age)
minimo <- min(mean_age)
maximo
```

```
## [1] 39.7
```

```
minimo
```

```
## [1] 24.1
```

```
relacion <- maximo / minimo
redondeado <- round(relacion, 2)

writeLines(paste("La relación entre el máximo y el mínimo es", redondeado))
```

```
## La relación entre el máximo y el mínimo es 1.65
```

- **EJERCICIO - 3** ***Referencias - https://www.rdocumentation.org/packages/base/versions/3.6.2/topics/scale***
- Normalizamos la matriz con scales, cargamos la libreria y creamos Xs, nos marca un error indicando **function (x, center = TRUE, scale = TRUE)** lo incluimos.

```
library(scales)
Xs <- scale(x, center = TRUE, scale = TRUE)
Xs
```

```
##              rent     inc_sal      inc_ret      inc_emp      inc_non
##  [1,]  0.11215097 -0.00395886 -0.931139719  2.418143014 -0.214973914
##  [2,] -0.08473303 -0.55542215  0.645412259 -1.550917436 -0.530200087
```

```
##  [3,]  0.21537779  0.44131517 -0.007907388 -0.167760007 -0.341064383
##  [4,] -0.64918611 -0.56595790 -0.712153958  0.644093267 -0.175570643
##  [5,] -2.27947972 -2.66395824 -1.775799086 -3.144555344  5.057183822
##  [6,] -1.53712508 -1.81416690 -0.840198788 -0.438377765  2.094057800
##  [7,]  1.72393277  1.23981404  0.982984995 -0.829270082  0.147536184
##  [8,]  0.98942212  1.12225726  0.258367657 -0.007393928  1.408440874
##  [9,]  0.97593047  0.26664354  1.488616571 -0.949544641  0.320910579
## [10,]  0.77794832  0.27579300  1.026636642 -0.438377765  0.005684406
## [11,]  1.07350963  0.91514662  1.108119716 -0.137691367  1.219305171
## [12,]  0.42010577 -0.03584336  1.236892074 -0.929498881 -0.916352148
## [13,]  1.02989708  0.93538635  0.627951601 -0.668904003  0.336671888
## [14,]  0.96008562  0.54279107  0.863670494 -0.157737127 -0.081002791
## [15,]  0.07449984  0.22505505  0.297654139  0.503772948 -0.017957557
## [16,]  0.08077503  0.66034784 -0.311286334  0.203086551 -0.742977753
## [17,] -0.34154506  0.25971212 -0.987159333  1.065054224 -0.798142334
## [18,]  0.75849524  0.22893665  2.109197484 -1.079842080  0.234223381
## [19,]  0.47234671 -0.26374825  1.600655798  0.082811992 -0.735097099
## [20,] -0.66801167 -0.64636230 -0.689600607  0.814482226  0.975004887
## [21,]  1.48108303  1.83868820  0.624313963 -0.508537924 -0.845426260
## [22,]  2.39835851  1.78351415  3.336536288 -2.823823186 -1.389191407
## [23,]  0.80587290  1.06625144  0.140508211  0.323361110 -0.979397383
## [24,]  0.51250790  1.15469628 -0.245081336  0.453658549 -1.302504210
## [25,]  0.48630900  1.29748341 -0.899856039 -0.087576967 -0.806022988
## [26,]  1.91579662  1.43666620  0.721075114 -1.089864960 -0.199212606
## [27,]  0.17192212  0.37837793  0.312932216 -0.418332005  0.060848987
## [28,] -0.55223447  0.21562833 -1.842004084  0.283269590  0.202700764
## [29,]  0.54482512  0.06507802  0.556653911  0.764367826 -0.356825692
## [30,]  1.13830094  0.53807770  1.193967955  0.744322066 -0.151928680
## [31,]  0.43234239  0.54694991 -0.301828477  0.032697592  0.573091517
## [32,]  0.97091032  1.18269919  0.276555843 -1.180070879 -0.262257840
## [33,]  1.17077503  1.35487551  0.842572198 -1.370505597 -0.301661112
## [34,] -0.13666020 -0.21716915  0.804013243 -1.069819200  0.975004887
## [35,] -1.11794258 -1.11575764 -0.549915337 -0.318103205 -0.703574482
## [36,] -0.92388242 -0.61253700 -1.046089056  0.714253427 -0.790261679
## [37,] -1.56191207 -1.66112129 -0.952238015 -0.298057446 -0.388348309
## [38,] -0.88152491 -0.67491972 -0.873665051  0.443635669  0.281507307
## [39,] -1.24940775 -1.21667903 -0.938414994  0.583955988 -0.632648593
## [40,] -1.22603268 -1.24135486 -0.757260659  0.654116147 -0.735097099
## [41,] -1.39640401 -1.60650174 -0.493895723 -0.237920166  0.510046282
## [42,] -1.43264322 -1.35807987 -1.216330479 -0.318103205 -0.813903642
## [43,] -0.88199555 -1.10854897  0.248909800  0.062766232 -0.530200087
## [44,] -0.63553758 -0.77750465  0.445342211  0.343406870 -0.538080741
## [45,] -0.85548288 -0.53213260 -1.071552517  0.363452629 -0.396228964
## [46,] -1.06868237 -0.70735874 -1.433133658  1.024962704 -0.813903642
## [47,]  0.54968839  1.04407092 -0.269817270  0.082811992  0.100252258
## [48,]  0.51078223  0.73160278  0.758906541  0.183040791  0.431239739
## [49,]  0.06508706  0.12718349 -1.156673228  2.809035331 -1.397072061
## [50,] -0.20427534 -0.17114456 -0.733979781  1.085099984  0.210581419
## [51,]  1.84535764  1.39369143  1.801453373 -1.671191995 -0.167689989
## [52,] -0.80355571 -0.68157388 -0.591384401  0.754344946  0.368194505
## [53,] -0.90348806 -0.89312063 -0.660499509  0.944779665 -0.033718865
## [54,] -0.01554909 -0.11513874  0.218353648  0.233155190 -0.427751581
## [55,] -1.15276987 -1.41713552 -0.552097919  0.624047507  1.471486109
## [56,]  0.69448833  1.53925112 -0.539002425  0.994894065 -0.932113457
```

```
## [57,] -0.23769071  0.07200944  0.013190907 -0.237920166  0.313029925
## [58,] -0.59945525 -0.29591001 -0.798729724  0.934756785  0.935601615
## [59,] -0.66440344 -0.48832606 -0.406592430  0.373475509 -0.214973914
## [60,] -0.40288501 -0.29008762 -0.212342601  0.062766232  0.714943295
## [61,] -0.89438904 -1.14847392  0.256185075  0.523818708  1.715786392
##           inc_oth        gini   dist8020   mean_age    perc_chil      per_ret
##  [1,]  0.79806710  2.39053026  1.7676480 -0.5671366 -1.512406683 -1.169959532
##  [2,]  0.35484048  1.21938116  2.5907512  0.7141721  0.414628464  1.302649642
##  [3,] -0.06529216  0.13392589 -0.4272940  0.1650398 -0.458559337  0.119178584
##  [4,] -0.51384816  0.33387817  0.6701770 -0.2010484 -0.368229565 -0.303489651
##  [5,] -1.27284043  0.81947658  0.1214415 -3.7887128  4.599907925 -2.268896942
##  [6,] -1.03079784 -0.80870632  0.1214415 -0.3840925  1.378146038  0.288245878
##  [7,]  2.32315748  0.56239507 -0.1529263  1.1534779 -0.037020399  1.281516230
##  [8,]  0.76653494  1.07655810  1.2189125  0.2748662 -0.428449413 -0.028755298
##  [9,]  1.45091592  0.59095969 -0.1529263  1.2999132 -0.428449413  1.535117171
## [10,]  1.15380309  0.67665352  0.9445447  1.3365220 -0.880098276  1.492850347
## [11,]  0.83404040 -0.38023714 -0.1529263  0.8606074 -0.488669261  1.239249406
## [12,]  0.58533609  0.76234736  0.9445447  1.0436515 -0.548889110  1.218115995
## [13,]  1.04543907  1.19081654  0.6701770  0.2748662 -0.247789868  0.182578819
## [14,]  1.33322549  0.44813662 -0.4272940  0.7873897 -0.669328806  0.584113643
## [15,] -0.35219035 -0.58018942 -0.4272940  0.6409544 -1.602736456 -0.176689180
## [16,] -0.60577993 -1.18004628 -1.2503973 -0.2010484 -0.037020399 -0.599357415
## [17,] -0.78164941 -0.26597869 -0.1529263 -1.2993131  0.565178085 -1.317893414
## [18,]  0.52848939  0.21961972  0.3958092  2.1419161 -1.120977669  2.591787758
## [19,]  0.82071696  1.61928573  1.7676480  1.0436515 -0.428449413  1.598517406
## [20,] -0.52495103  0.01966744  0.3958092 -0.2376573  0.083419298 -0.303489651
## [21,]  0.93707505 -0.29454330 -0.9760295  0.0552133  0.053309374 -0.134422357
## [22,]  2.09976771 -0.23741407 -0.4272940  1.4463485 -1.030647896  1.196982583
## [23,]  0.52893350  0.21961972 -0.4272940  0.2016486  0.324298692  0.267112466
## [24,] -0.19541781 -1.40856318 -1.2503973 -0.2376573  0.836167403 -0.409156709
## [25,] -0.10259781 -1.15148167 -1.2503973 -1.2993131  1.588915507 -1.423560473
## [26,]  2.74151367  2.13344875  0.9445447  0.1650398 -0.247789868  0.161445408
## [27,] -0.29445542 -0.12315562 -0.1529263  0.2382574 -0.006910474  0.267112466
## [28,] -0.80829630 -0.49449558 -0.7016618 -2.8368835  2.311553687 -2.332297178
## [29,]  1.08452117  1.59072112  2.0420157  0.2382574 -0.910208200  0.309379290
## [30,]  1.60768846  0.73378275  0.1214415  0.9704338 -1.482296759  0.816581172
## [31,]  0.49828958  0.59095969  1.2189125 -0.2742661 -0.609108958 -0.514823768
## [32,]  0.75321150 -1.09435244 -0.9760295  0.3846927 -0.338119640 -0.007621886
## [33,]  0.70746767 -1.03722322 -0.9760295  0.0552133  0.294188767 -0.218956004
## [34,] -0.53738624  0.24818434  0.6701770  0.6409544  0.053309374  0.816581172
## [35,] -0.98771870  0.50526585  0.3958092 -0.1644396 -0.247789868 -0.049888710
## [36,] -0.98283344 -0.26597869 -0.1529263 -0.7501807 -0.187570019 -0.937492003
## [37,] -1.14360301 -0.23741407 -0.1529263 -0.8233984  0.986717024 -0.240089415
## [38,] -0.91666032 -0.06602640 -0.1529263 -0.9332248  0.474848312 -0.789558121
## [39,] -1.00503918  0.13392589 -0.4272940 -1.0796601  0.866277327 -0.641624238
## [40,] -1.00725975  0.04823205  0.6701770 -0.7867896  0.595288009 -0.578224003
## [41,] -1.09563861 -0.69444787 -0.1529263 -0.3108749  0.595288009  0.309379290
## [42,] -1.07742990  0.30531356  0.6701770 -1.4457483  0.806057478 -1.085425885
## [43,] -0.84515783  0.24818434  0.6701770  0.3480839 -0.488669261  0.816581172
## [44,] -0.81007276  0.01966744 -0.1529263  0.6409544 -0.639218882  0.710914113
## [45,] -0.90822214 -0.66588326 -0.4272940 -0.4939190  0.233968919 -1.022025650
## [46,] -1.01658616 -0.43736636  0.3958092 -1.4457483  0.986717024 -1.508094120
## [47,]  0.03951894 -0.60875403 -0.9760295 -0.5671366  0.113529222 -0.916358591
## [48,] -0.22162059 -0.20884946 -0.1529263  0.4579103 -0.849988351  0.288245878
```

```
## [49,]  0.64129456  1.61928573  1.7676480 -0.6403543 -1.452186835 -1.423560473
## [50,]  0.08393043  2.27627181  1.7676480 -0.1278308 -0.458559337 -0.620490827
## [51,]  1.97585967  0.16249050 -0.7016618  1.1534779 -0.368229565  1.154715760
## [52,] -0.87624587  0.07679666  0.1214415  0.3114750 -0.669328806  0.098045173
## [53,] -0.76388482 -0.95152938 -0.1529263 -0.4207014  0.384518540 -0.493690356
## [54,]  0.02086612 -0.86583555 -1.2503973  0.6043456  0.565178085  0.626380466
## [55,] -0.76610539  0.67665352  0.9445447  0.1284309  0.655507858  0.499579996
## [56,] -0.16122097 -1.69420930 -1.7991328 -0.4573102  0.504958237 -1.212226355
## [57,] -0.80296692 -1.83703237 -1.7991328  0.3480839 -0.067130323 -0.282356239
## [58,] -0.82872558 -1.69420930 -1.5247650 -0.4573102  0.956607099 -0.345756474
## [59,] -0.85492836 -2.06554926 -1.7991328  0.1284309 -0.127350171 -0.240089415
## [60,] -0.58890357 -0.92296477 -0.7016618  0.2016486  0.053309374 -0.197822592
## [61,] -0.96817765 -1.37999857 -1.2503973  1.7758279 -1.843615849  2.021185641
##         home_size
##  [1,] -1.54065428
##  [2,]  0.66528253
##  [3,] -0.87537175
##  [4,] -0.91038662
##  [5,]  2.97626394
##  [6,]  0.63026766
##  [7,] -0.70029740
##  [8,] -1.08546097
##  [9,] -0.73531227
## [10,] -0.91038662
## [11,] -1.22552045
## [12,] -0.80534201
## [13,] -0.17507435
## [14,] -0.49020818
## [15,] -0.38516357
## [16,] -0.73531227
## [17,] -0.35014870
## [18,] -0.80534201
## [19,] -0.98041636
## [20,]  0.21008922
## [21,]  0.59525279
## [22,]  1.40059480
## [23,]  1.50563941
## [24,] -0.17507435
## [25,]  0.17507435
## [26,]  0.80534201
## [27,] -0.42017844
## [28,]  0.07002974
## [29,] -1.36557993
## [30,] -1.61068401
## [31,] -0.87537175
## [32,] -0.35014870
## [33,] -0.28011896
## [34,]  0.03501487
## [35,]  0.28011896
## [36,]  0.42017844
## [37,]  1.82077323
## [38,]  1.19050558
## [39,]  1.22552045
## [40,]  1.75074349
```

```
## [41,]  1.50563941
## [42,]  1.96083271
## [43,] -0.14005948
## [44,]  0.35014870
## [45,]  0.28011896
## [46,]  0.77032714
## [47,] -0.42017844
## [48,] -0.17507435
## [49,] -1.78575836
## [50,] -1.22552045
## [51,]  0.84035688
## [52,] -1.01543123
## [53,]  0.07002974
## [54,] -0.14005948
## [55,]  0.84035688
## [56,]  0.42017844
## [57,]  0.28011896
## [58,]  0.31513383
## [59,]  0.49020818
## [60,]  0.49020818
## [61,] -1.68071375
## attr(,"scaled:center")
##         rent      inc_sal      inc_ret      inc_emp      inc_non      inc_oth
## 19351.114754 11524.278689  3641.868852   656.737705   617.278689  2910.016393
##         gini     dist8020     mean_age    perc_chil      per_ret    home_size
##    31.331148     2.755738    41.049180    18.822951    17.136066     2.610000
## attr(,"scaled:scale")
##         rent      inc_sal      inc_ret      inc_emp      inc_non      inc_oth
## 6374.3119344 3606.7677549 1374.5185882   99.7717231  126.8930168 2251.6698285
##         gini     dist8020     mean_age    perc_chil      per_ret    home_size
##    3.5008352    0.3644743    2.7315821    3.3211641    4.7318436    0.2855930
```

- La matriz de covarianzas se calcula de la siguiente manera como se indica en los apuntes:
- Nos da un error para representar el .plot y es debido a que no tenemos instalado la libreria fields.**install.packages("fields")**
- Para invertir la matriz - image.plot(CXs[,nrow(CXs):1])

```
library(fields)
```

```
## Loading required package: spam
```

```
## Spam version 2.9-1 (2022-08-07) is loaded.
## Type 'help( Spam)' or 'demo( spam)' for a short introduction
## and overview of this package.
## Help for individual functions is also obtained by adding the
## suffix '.spam' to the function name, e.g. 'help( chol.spam)'.

##
## Attaching package: 'spam'

## The following objects are masked from 'package:base':
##
##     backsolve, forwardsolve
```

```
## Loading required package: viridis

## Loading required package: viridisLite

##
## Attaching package: 'viridis'

## The following object is masked from 'package:scales':
##
##     viridis_pal

##
## Try help(fields) to get started.
```

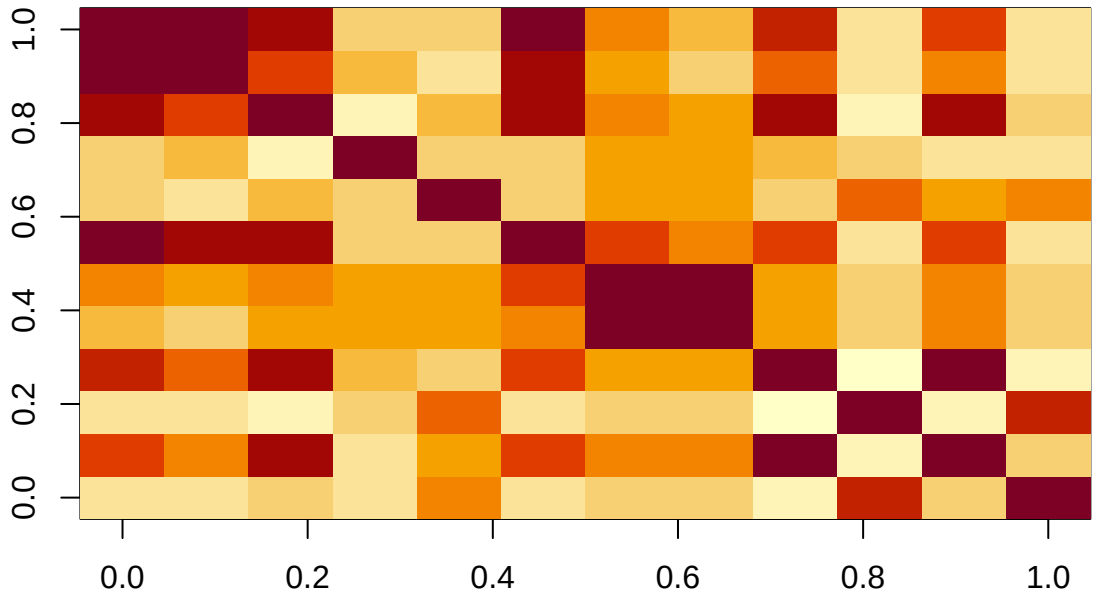
```
CXs <- cov(Xs)

CXs
```

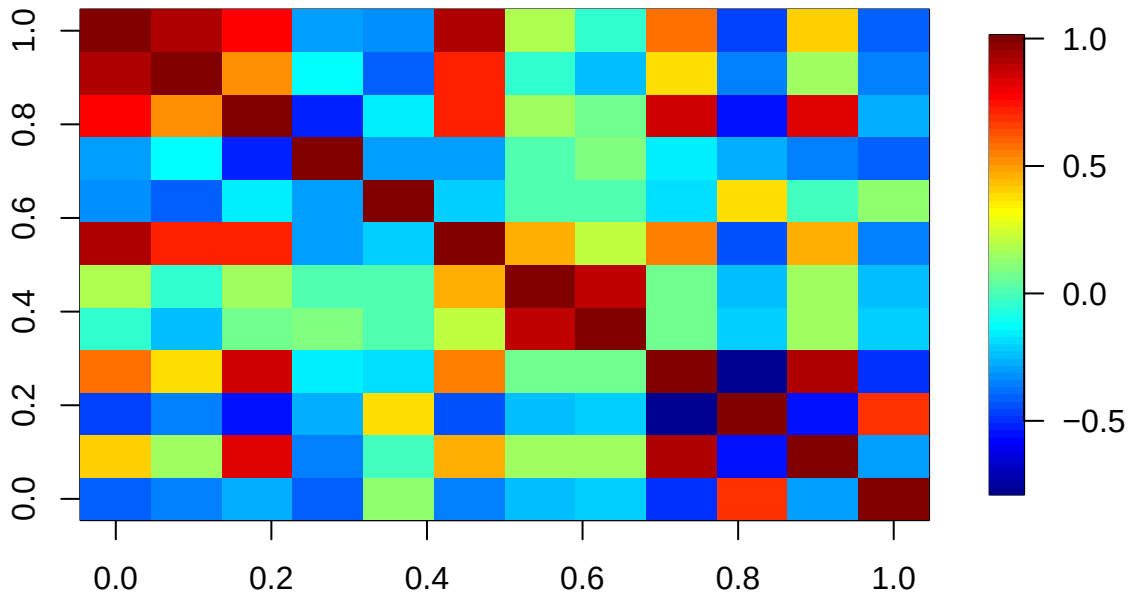
```
##                 rent     inc_sal     inc_ret     inc_emp     inc_non
## rent       1.00000000  0.92673026  0.76122489 -0.28365656 -0.31599008
## inc_sal    0.92673026  1.00000000  0.52982883 -0.13254741 -0.39950272
## inc_ret    0.76122489  0.52982883  1.00000000 -0.51158121 -0.16092694
## inc_emp   -0.28365656 -0.13254741 -0.51158121  1.00000000 -0.29219282
## inc_non   -0.31599008 -0.39950272 -0.16092694 -0.29219282  1.00000000
## inc_oth    0.91309520  0.72773849  0.72831587 -0.30487998 -0.20198487
## gini       0.18017222 -0.03060094  0.16273694  0.01873346  0.02066379
## dist8020  -0.04145366 -0.23533662  0.05805812  0.09633633  0.02290216
## mean_age   0.57540695  0.37060666  0.84615522 -0.16870717 -0.19666356
## perc_chil -0.47169903 -0.35485651 -0.55011339 -0.27748580  0.37766463
## per_ret    0.41627073  0.14320960  0.83398303 -0.35580857 -0.00395581
## home_size -0.39501073 -0.36463033 -0.27233745 -0.39697106  0.12947569
##               inc_oth        gini    dist8020    mean_age  perc_chil     per_ret
## rent       0.9130952  0.18017222 -0.04145366  0.57540695 -0.4716990  0.41627073
## inc_sal    0.7277385 -0.03060094 -0.23533662  0.37060666 -0.3548565  0.14320960
## inc_ret    0.7283159  0.16273694  0.05805812  0.84615522 -0.5501134  0.83398303
## inc_emp   -0.3048800  0.01873346  0.09633633 -0.16870717 -0.2774858 -0.35580857
## inc_non   -0.2019849  0.02066379  0.02290216 -0.19666356  0.3776646 -0.00395581
## inc_oth    1.0000000  0.45736948  0.21856466  0.53895303 -0.4420784  0.45691555
## gini       0.4573695  1.00000000  0.87938833  0.07983447 -0.2432927  0.14512320
## dist8020   0.2185647  0.87938833  1.00000000  0.05947531 -0.2204091  0.14831518
## mean_age   0.5389530  0.07983447  0.05947531  1.00000000 -0.7753111  0.91512413
## perc_chil -0.4420784 -0.24329266 -0.22040906 -0.77531110  1.0000000 -0.55090986
## per_ret    0.4569156  0.14512320  0.14831518  0.91512413 -0.5509099  1.00000000
## home_size -0.3588892 -0.22597540 -0.21135293 -0.48257496  0.6871720 -0.30387429
##            home_size
## rent      -0.3950107
## inc_sal   -0.3646303
## inc_ret   -0.2723374
## inc_emp   -0.3969711
## inc_non    0.1294757
## inc_oth   -0.3588892
## gini      -0.2259754
```

```
## dist8020  -0.2113529
## mean_age  -0.4825750
## perc_chil  0.6871720
## per_ret   -0.3038743
## home_size  1.0000000
```

```
image(CXs[,nrow(CXs):1])
```




```
image.plot(CXs[,nrow(CXs):1],legend.only = FALSE)
```



Para obtener el valor absoluto - https://openwebinars.net/blog/conoce-las-funciones-predefinidas-en-r-mas-usadas/

```
Absmax = abs(CXs-diag(diag(CXs)))
Absoluto <- abs(CXs)
MaxAbsol <- max(abs(Absoluto))
writeLines(paste("El valor máximo es de ", MaxAbsol))
```

```
## El valor máximo es de  1
```

```
MinAbsol <- min(Absoluto)
writeLines(paste("El valor mínimo es", MinAbsol))
```

```
## El valor mínimo es 0.00395581045248607
```

```
MaximosAbsoluto <- which(Absmax  == max(Absmax ), arr.ind = TRUE, useNames = TRUE)
MaximosAbsoluto
```

```
##         row col
## inc_sal   2   1
## rent      1   2
```

```r
Maxinombres <- colnames(Absoluto)[MaximosAbsoluto[,2]]
Maxinombres
```

```
## [1] "rent"    "inc_sal"
```

```r
MinimoAbsoluto <- which(Absoluto == min(Absoluto), arr.ind = TRUE, useNames = TRUE)
MinimoAbsoluto
```

```
##         row col
## per_ret  11   5
## inc_non   5  11
```

```r
Minimonombres <- colnames(Absoluto)[MinimoAbsoluto[,2]]
Minimonombres
```

```
## [1] "inc_non" "per_ret"
```

- **EJERCICIO - 4 Referencia - https://rpubs.com/aaronsc32/singular-value-decomposition-r *https://rpubs.com/Joaquin_AR/287787*** Se realiza la descomposicón en componentes (SVD) y la descomposición de componentes principales PCA Se representa Pca

```r
A.svd <- svd(CXs)
writeLines(paste("La descomposición en componentes principales (SVD) de la matriz de
covarianzas es ", A.svd))
```

```
## La descomposición en componentes principales (SVD) de la matriz de
## covarianzas es  c(5.2184833480441, 2.12534058200506, 1.83652075178474, 1.42519142406365, 0.789519644
## La descomposición en componentes principales (SVD) de la matriz de
## covarianzas es  c(-0.383121333597568, -0.301861808509468, -0.39322498672812, 0.0958926034088017, 0.1
## -0.196258955989622, 0.527213413242267, -0.444850211815632, -0.0877994871489636, -0.301655353930872,
## 0.777800206560312, 0.0974701341744418, -0.0498279948045908, -0.152274643797746, -0.0782813989050987,
## -0.00537042955683046, -0.0400327357081934, 0.0878186400770634, 0.709766722469097, 0.451285251103635,
## -0.0693345699423898, 0.111197149251893, 0.097600379695527, -0.0163862547706678, 0.0346803850636784,
## 0.286385454739905, -0.602003809704286, -0.0446665728032373, 0.818732046164689, -0.462751079977998, -
## La descomposición en componentes principales (SVD) de la matriz de
## covarianzas es  c(-0.383121333597568, -0.301861808509468, -0.39322498672812, 0.0958926034088017, 0.1
## -0.196258955989622, 0.527213413242267, -0.444850211815632, -0.0877994871489635, -0.301655353930872,
## 0.777800206560313, 0.0974701341744416, -0.0498279948045912, -0.152274643797746, -0.0782813989050987,
## -0.00537042955683041, -0.0400327357081933, 0.0878186400770632, 0.709766722469097, 0.451285251103635,
## -0.0693345699423898, 0.111197149251893, 0.097600379695526, -0.016386254770668, 0.0346803850636782, -
## 0.286385454739905, -0.602003809704285, -0.0446665728032372, 0.818732046164689, -0.462751079977998, -
```

```r
pca <- prcomp(CXs, scale = TRUE)
pca$center
```

```
##        rent     inc_sal     inc_ret     inc_emp     inc_non     inc_oth
##  0.27209085  0.18171994  0.28544533 -0.13406340 -0.00170921  0.31109332
##        gini    dist8020    mean_age   perc_chil     per_ret   home_size
##  0.20367941  0.14787398  0.23019158 -0.15177660  0.23703274 -0.09958073
```

```
pca$scale
```

```
##      rent   inc_sal   inc_ret   inc_emp   inc_non   inc_oth      gini  dist8020
## 0.5625456 0.5154974 0.5655764 0.3956210 0.3826377 0.5175665 0.3911565 0.3986062
##  mean_age perc_chil   per_ret home_size
## 0.5710852 0.5455824 0.5069376 0.4710212
```

```
pca$rotation
```

```
##                   PC1         PC2         PC3         PC4         PC5
## rent      -0.35100167 -0.14230458  0.05220768  0.26391096 -0.18516047
## inc_sal   -0.31638931 -0.19961727  0.22704272  0.35336521 -0.27336381
## inc_ret   -0.35084900 -0.14997244 -0.18897178 -0.06871054  0.11329205
## inc_emp    0.07368784  0.42410571  0.59154527 -0.24139683 -0.10574433
## inc_non    0.24800885 -0.04619744 -0.48759531 -0.27977120 -0.70968613
## inc_oth   -0.35478276 -0.03159365 -0.07420498  0.31333013 -0.17501454
## gini      -0.13077457  0.53762738 -0.29673849  0.33435832 -0.02053500
## dist8020  -0.05889222  0.59488956 -0.30351710  0.13347547  0.14579720
## mean_age  -0.35191846 -0.05063484 -0.08416606 -0.34678452  0.13346110
## perc_chil  0.34494718 -0.16075214 -0.11629273  0.28087600 -0.04503922
## per_ret   -0.32069017 -0.06497889 -0.27809618 -0.41603555  0.22558106
## home_size  0.30747990 -0.24258694 -0.19675487  0.25336767  0.49018431
##                   PC6         PC7         PC8          PC9         PC10
## rent      -0.053773095  0.04503350  0.08895514  0.015932286  0.17208138
## inc_sal    0.060969553  0.33909535 -0.11127257 -0.357568776  0.39259627
## inc_ret   -0.009133726  0.08633491 -0.16095280  0.781145650  0.22294389
## inc_emp   -0.277235024 -0.22503180  0.18378462  0.108066520  0.40114289
## inc_non   -0.258196053  0.10272342  0.09104178 -0.009434785  0.15306603
## inc_oth   -0.236149208 -0.45610198  0.55002124  0.091603765 -0.26425406
## gini      -0.244202108 -0.27842152 -0.58032129 -0.117246386  0.01491597
## dist8020   0.297217157  0.41741957  0.47766399  0.001156040  0.12770797
## mean_age  -0.086334969  0.01645320  0.05557989 -0.374658741 -0.32908942
## perc_chil  0.518762897 -0.47539465  0.07588050 -0.020922901  0.21190439
## per_ret    0.136365350 -0.34448592  0.03880256 -0.273258654  0.51305046
## home_size -0.592351203  0.09804694  0.18418071 -0.110987771  0.28181777
##                  PC11          PC12
## rent       0.06141675 -0.8387000438
## inc_sal    0.08446641  0.4343975658
## inc_ret    0.27289012  0.1825985167
## inc_emp    0.24773433  0.0090941632
## inc_non    0.08673100  0.0116398740
## inc_oth   -0.14127504  0.2726167076
## gini       0.07542282  0.0002956350
## dist8020   0.06627372 -0.0002733527
## mean_age   0.68480042 -0.0001312617
## perc_chil  0.46315457  0.0002572786
## per_ret   -0.33906117 -0.0006676546
## home_size  0.13139304 -0.0001481856
```

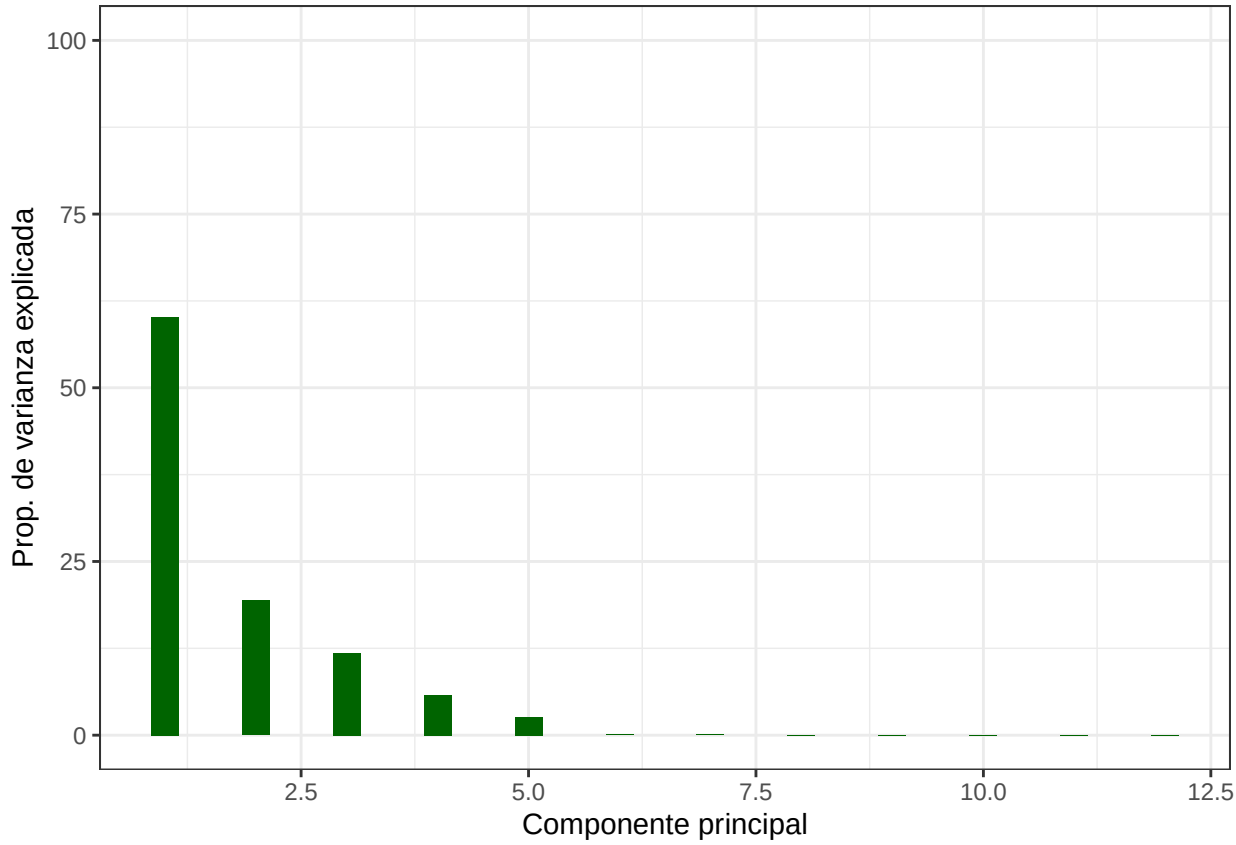
```
library(ggplot2)
pca$sdev^2
```

```
##  [1] 7.222877e+00 2.322687e+00 1.421236e+00 6.953089e-01 3.123226e-01
##  [6] 1.590081e-02 6.769965e-03 1.852193e-03 8.252834e-04 2.152906e-04
## [11] 5.202595e-06 3.375466e-34
```

```
prop_varianza <- pca$sdev^2 / sum(pca$sdev^2) *100

df <- data.frame(prop_varianza, pc = 1:length(prop_varianza))

ggplot(data = df, aes(x = pc, y = prop_varianza)) +
  geom_col(width = 0.3, fill = "#006400") +
  scale_y_continuous(limits = c(0, 100)) +
  theme_bw() +
  labs(x = "Componente principal", y = "Prop. de varianza explicada")
```



- **EJERCICIO - 5** ***Referencia - https://rpubs.com/JairoAyala/574796*** Reducimos la dimensionalidad de la matriz.

```
pca <- prcomp(Xs, scale = TRUE)
prop_varianza <- pca$sdev^2 / sum(pca$sdev^2)
prop_varianza_acum <- cumsum(prop_varianza)
L <- which(prop_varianza_acum >= 0.75)[1]
L
```

```
## [1] 3
```

- **EJERCICIO - 6** ***Referencias - https://rpubs.com/Joaquin_AR/287787*** El fallo del intento anterior fue no contabilizar **números absolutos** ***

```
pcarent <- pca$rotation[,1]
pcarent
```

```
##        rent     inc_sal     inc_ret     inc_emp     inc_non     inc_oth
## -0.38312133 -0.30186181 -0.39322499  0.09589260  0.13729850 -0.37361445
##        gini    dist8020    mean_age   perc_chil     per_ret   home_size
## -0.13487528 -0.07551688 -0.37797355  0.32816462 -0.32429938  0.24539219
```

```
Maxpcainc_ret<- max(abs(pcarent))
column_index <- which.max(abs(pcarent))
column_name <- colnames(Xs)[column_index]
writeLines(paste("Variables contribuyen en mayor",column_name,Maxpcainc_ret))
```

```
## Variables contribuyen en mayor inc_ret 0.39322498672812
```

```
Minpcainc_ret<- min(abs(pcarent))

column_index <- which.min(abs(pcarent))
column_name <- colnames(Xs)[column_index]
writeLines(paste("Variables contribuyen menor",column_name,Minpcainc_ret))
```

```
## Variables contribuyen menor dist8020 0.0755168790570722
```

- **EJERCICIO - 7** Se selecciona la columna 1, se indica cual es el valor máximo y se busca en el df original del comienzo donde no se ha eliminado la primera columna donde está el id.

```
pcarent <- pca$x[,1]
Maxpcainc_ret<- which(pcarent == max(pcarent))
Minpcainc_ret<- which(pcarent == min(pcarent))

Censal_max <- var_df[, 1][Maxpcainc_ret]
Censal_min <- var_df[, 1][Minpcainc_ret]

Censal_max
```

```
## [1] 5
```

```
Censal_min
```

```
## [1] 22
```

- **EJERCICIO - 8** ***Referencias https://r-coder.com/varianza-desviacion-tipica-r/ https://programmerclick.com/article/21811287234/***

```r
pcaL <- prcomp(Xs, rank = 3)
pca2 <- pca$x

reconstructed <- pcaL$x %*% t(pcaL$rotation[, 1:3])
error_residual <- Xs - reconstructed

sd(x=error_residual)
```

```
## [1] 0.4810774
```

```r
sd
```

```
## function (x, na.rm = FALSE)
## sqrt(var(if (is.vector(x) || is.factor(x)) x else as.double(x),
##     na.rm = na.rm))
## <bytecode: 0x000002ba30cb7ee0>
## <environment: namespace:stats>
```

MANUEL ROJAS GARCÍA