# PR1-P2023

Alumno: Borja Villena Pardo

**Práctica – Borja Villena Pardo – Intento 2**

* **La resolución de la práctica (memoria técnica detallada). Importante: debéis especificar a que intento de la tabla corresponde.**
* **El código en R.**
* **Las imágenes y/o figuras que se os pidan.**

***Datos:***

* **id:** identificador de la sección censal.
* **1- rent:** renta bruta por persona.
* **2- inc\_sal:** ingresos provenientes del salario.
* **3- inc\_ret:** ingresos provenientes de pensiones de jubilación.
* **4- inc\_emp:** ingresos provenientes de prestaciones del paro.
* **5- inc\_non:** ingresos provenientes de otros tipos de prestaciones.
* **6- inc\_oth:** otros ingresos.
* **7- gini:** coeficiente de Gini que mide la desigualdad.
* **8- dist8020:** relación de renta entre el percentil 80 (P80) y el percentil 20 (P20) − P80/P20.
* **9- mean\_age:** edad media de la población.
* **10- perc\_chil:** porcentaje de población menor de 18 años.
* **11- per\_ret:** porcentaje de población mayor de 65 años.
* **12- home\_size:** tamaño medio del hogar (m2).

**Pregunta 0.** Los resultados del informe corresponden al primer intento:

***Respuesta:***

* V = gini
* C = 1
* Copiamos código R usado para conseguir la respuesta: NA

**Pregunta 1*.*** [10 %] Para empezar, generar la matriz X a partir del *dataframe* **var\_df** utilizando, por ejemplo, la instrucción **as.matrix.** Aseguraros de eliminar el valor de la primera columna ('id') ya que no proporciona ninguna información relevante. Comprobar, también, que X es una matriz y no un *dataframe* utilizando la siguiente instrucción:

1 > class (X)

2 [1] " matrix " " array "

Responder: ¿cuántas secciones censales tiene la ciudad?

***Respuesta:***

* La ciudad tiene **61** secciones censales.
* Copiamos código R usado para conseguir la respuesta:

> ***#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\\variables.csv")**

>

> ***#Imprimimos archivo para ver su contenido***

**> fix(var\_df)**

**> var\_df**

>

> ***#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**

>

> ***#Convertimos var\_df en una matriz***

**> var\_matrix <- as.matrix(var\_df)**

>

> ***#Comprobamos que ahora 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.*** [10 %] Como alcaldables, os interesa tener una primera impresión general de las variables medidas y explorar los datos en crudo. Una de las características interesantes a estudiar es la razón (*M/m*) entre el valor máximo (*M)* y el mínimo (*m*) de una variable. Calcular la razón de la variable V.

***Respuesta:***

* La razón de la variable V es igual a **1.65**
* Copiamos código R usado para conseguir la respuesta:

***> #En este intento, la variable V es igual a la variable 'gini'. Para calcular***

***> #la razón de 'gini' vamos a detectar los valores máximo y mínimo de la variable.***

> ***#Asignamos a V la columna 'gini' de la matriz var\_df***

**> V <- var\_matrix[,7]**

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18

39.7 35.6 31.8 32.5 34.2 28.5 33.3 35.1 33.4 33.7 30.0 34.0 35.5 32.9 29.3 27.2 30.4 32.1

19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36

37.0 31.4 30.3 30.5 32.1 26.4 27.3 38.8 30.9 29.6 36.9 33.9 33.4 27.5 27.7 32.2 33.1 30.4

37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54

30.5 31.1 31.8 31.5 28.9 32.4 32.2 31.4 29.0 29.8 29.2 30.6 37.0 39.3 31.9 31.6 28.0 28.3

55 56 57 58 59 60 61

33.7 25.4 24.9 25.4 24.1 28.1 26.5

***> #Definimos máximo como el valor ‘máximo’ de V, y definimos ‘mínimo’***

***> #como el valor mínimo de V***

**> maximo <- max(V)**

**> minimo <- min(V)**

**> maximo**

[1] 39.7

**> minimo**

[1] 24.1

***> #Calculamos la razón entre el valor Máximo y el valor Mínimo de V (M/m).***

***>#Guardamos valor en ‘razon’***

**> razon <- maximo/minimo**

**> razon**

[1] 1.647303

***> #Redondeamos resultado a dos decimales***

**> round(razon, digits=2)**

[1] 1.65

***Pregunta 3.*** [15 %] Para poder realizar el análisis de componentes principales debéis, inicialmente, normalizar los datos y guardarlos a la variable ***Xs***, como se muestra a la Sección 2.1 de los apuntes del módulo. Una vez normalizados, calcular la matriz de covarianzas de los datos; guardarla en la variable ***CXs*** y adjuntarla 1 como una imagen. Finalmente, indicar cuales son el par de variables (distintas) con mayor covarianza (en valor absoluto) y el par con menor covarianza (en valor absoluto).

***Respuesta:***

* Los pares con MAYOR varianza en valores absolutos son: {1, 2}.
* Los pares con MENOR varianza en valores absolutos son: {5, 11}
* Copiamos código R usado para conseguir la respuesta:

***> #Calculamos la matriz de datos normalizada y la guardamos en la variable Xs***

**> Xs <- as.matrix(scale(var\_matrix, center = TRUE, scale = TRUE))**

**> Xs**

rent inc\_sal inc\_ret inc\_emp inc\_non inc\_oth gini

1 0.11215097 -0.00395886 -0.931139719 2.418143014 -0.214973914 0.79806710 2.39053026

2 -0.08473303 -0.55542215 0.645412259 -1.550917436 -0.530200087 0.35484048 1.21938116

3 0.21537779 0.44131517 -0.007907388 -0.167760007 -0.341064383 -0.06529216 0.13392589

4 -0.64918611 -0.56595790 -0.712153958 0.644093267 -0.175570643 -0.51384816 0.33387817

5 -2.27947972 -2.66395824 -1.775799086 -3.144555344 5.057183822 -1.27284043 0.81947658

6 -1.53712508 -1.81416690 -0.840198788 -0.438377765 2.094057800 -1.03079784 -0.80870632

7 1.72393277 1.23981404 0.982984995 -0.829270082 0.147536184 2.32315748 0.56239507

8 0.98942212 1.12225726 0.258367657 -0.007393928 1.408440874 0.76653494 1.07655810

9 0.97593047 0.26664354 1.488616571 -0.949544641 0.320910579 1.45091592 0.59095969

10 0.77794832 0.27579300 1.026636642 -0.438377765 0.005684406 1.15380309 0.67665352

11 1.07350963 0.91514662 1.108119716 -0.137691367 1.219305171 0.83404040 -0.38023714

12 0.42010577 -0.03584336 1.236892074 -0.929498881 -0.916352148 0.58533609 0.76234736

13 1.02989708 0.93538635 0.627951601 -0.668904003 0.336671888 1.04543907 1.19081654

14 0.96008562 0.54279107 0.863670494 -0.157737127 -0.081002791 1.33322549 0.44813662

15 0.07449984 0.22505505 0.297654139 0.503772948 -0.017957557 -0.35219035 -0.58018942

16 0.08077503 0.66034784 -0.311286334 0.203086551 -0.742977753 -0.60577993 -1.18004628

17 -0.34154506 0.25971212 -0.987159333 1.065054224 -0.798142334 -0.78164941 -0.26597869

18 0.75849524 0.22893665 2.109197484 -1.079842080 0.234223381 0.52848939 0.21961972

19 0.47234671 -0.26374825 1.600655798 0.082811992 -0.735097099 0.82071696 1.61928573

20 -0.66801167 -0.64636230 -0.689600607 0.814482226 0.975004887 -0.52495103 0.01966744

21 1.48108303 1.83868820 0.624313963 -0.508537924 -0.845426260 0.93707505 -0.29454330

22 2.39835851 1.78351415 3.336536288 -2.823823186 -1.389191407 2.09976771 -0.23741407

23 0.80587290 1.06625144 0.140508211 0.323361110 -0.979397383 0.52893350 0.21961972

24 0.51250790 1.15469628 -0.245081336 0.453658549 -1.302504210 -0.19541781 -1.40856318

25 0.48630900 1.29748341 -0.899856039 -0.087576967 -0.806022988 -0.10259781 -1.15148167

26 1.91579662 1.43666620 0.721075114 -1.089864960 -0.199212606 2.74151367 2.13344875

27 0.17192212 0.37837793 0.312932216 -0.418332005 0.060848987 -0.29445542 -0.12315562

28 -0.55223447 0.21562833 -1.842004084 0.283269590 0.202700764 -0.80829630 -0.49449558

29 0.54482512 0.06507802 0.556653911 0.764367826 -0.356825692 1.08452117 1.59072112

30 1.13830094 0.53807770 1.193967955 0.744322066 -0.151928680 1.60768846 0.73378275

31 0.43234239 0.54694991 -0.301828477 0.032697592 0.573091517 0.49828958 0.59095969

32 0.97091032 1.18269919 0.276555843 -1.180070879 -0.262257840 0.75321150 -1.09435244

33 1.17077503 1.35487551 0.842572198 -1.370505597 -0.301661112 0.70746767 -1.03722322

34 -0.13666020 -0.21716915 0.804013243 -1.069819200 0.975004887 -0.53738624 0.24818434

35 -1.11794258 -1.11575764 -0.549915337 -0.318103205 -0.703574482 -0.98771870 0.50526585

36 -0.92388242 -0.61253700 -1.046089056 0.714253427 -0.790261679 -0.98283344 -0.26597869

37 -1.56191207 -1.66112129 -0.952238015 -0.298057446 -0.388348309 -1.14360301 -0.23741407

38 -0.88152491 -0.67491972 -0.873665051 0.443635669 0.281507307 -0.91666032 -0.06602640

39 -1.24940775 -1.21667903 -0.938414994 0.583955988 -0.632648593 -1.00503918 0.13392589

40 -1.22603268 -1.24135486 -0.757260659 0.654116147 -0.735097099 -1.00725975 0.04823205

41 -1.39640401 -1.60650174 -0.493895723 -0.237920166 0.510046282 -1.09563861 -0.69444787

42 -1.43264322 -1.35807987 -1.216330479 -0.318103205 -0.813903642 -1.07742990 0.30531356

43 -0.88199555 -1.10854897 0.248909800 0.062766232 -0.530200087 -0.84515783 0.24818434

44 -0.63553758 -0.77750465 0.445342211 0.343406870 -0.538080741 -0.81007276 0.01966744

45 -0.85548288 -0.53213260 -1.071552517 0.363452629 -0.396228964 -0.90822214 -0.66588326

46 -1.06868237 -0.70735874 -1.433133658 1.024962704 -0.813903642 -1.01658616 -0.43736636

47 0.54968839 1.04407092 -0.269817270 0.082811992 0.100252258 0.03951894 -0.60875403

48 0.51078223 0.73160278 0.758906541 0.183040791 0.431239739 -0.22162059 -0.20884946

49 0.06508706 0.12718349 -1.156673228 2.809035331 -1.397072061 0.64129456 1.61928573

50 -0.20427534 -0.17114456 -0.733979781 1.085099984 0.210581419 0.08393043 2.27627181

51 1.84535764 1.39369143 1.801453373 -1.671191995 -0.167689989 1.97585967 0.16249050

52 -0.80355571 -0.68157388 -0.591384401 0.754344946 0.368194505 -0.87624587 0.07679666

53 -0.90348806 -0.89312063 -0.660499509 0.944779665 -0.033718865 -0.76388482 -0.95152938

54 -0.01554909 -0.11513874 0.218353648 0.233155190 -0.427751581 0.02086612 -0.86583555

55 -1.15276987 -1.41713552 -0.552097919 0.624047507 1.471486109 -0.76610539 0.67665352

56 0.69448833 1.53925112 -0.539002425 0.994894065 -0.932113457 -0.16122097 -1.69420930

57 -0.23769071 0.07200944 0.013190907 -0.237920166 0.313029925 -0.80296692 -1.83703237

58 -0.59945525 -0.29591001 -0.798729724 0.934756785 0.935601615 -0.82872558 -1.69420930

59 -0.66440344 -0.48832606 -0.406592430 0.373475509 -0.214973914 -0.85492836 -2.06554926

60 -0.40288501 -0.29008762 -0.212342601 0.062766232 0.714943295 -0.58890357 -0.92296477

61 -0.89438904 -1.14847392 0.256185075 0.523818708 1.715786392 -0.96817765 -1.37999857

dist8020 mean\_age perc\_chil per\_ret home\_size

1 1.7676480 -0.5671366 -1.512406683 -1.169959532 -1.54065428

2 2.5907512 0.7141721 0.414628464 1.302649642 0.66528253

3 -0.4272940 0.1650398 -0.458559337 0.119178584 -0.87537175

4 0.6701770 -0.2010484 -0.368229565 -0.303489651 -0.91038662

5 0.1214415 -3.7887128 4.599907925 -2.268896942 2.97626394

6 0.1214415 -0.3840925 1.378146038 0.288245878 0.63026766

7 -0.1529263 1.1534779 -0.037020399 1.281516230 -0.70029740

8 1.2189125 0.2748662 -0.428449413 -0.028755298 -1.08546097

9 -0.1529263 1.2999132 -0.428449413 1.535117171 -0.73531227

10 0.9445447 1.3365220 -0.880098276 1.492850347 -0.91038662

11 -0.1529263 0.8606074 -0.488669261 1.239249406 -1.22552045

12 0.9445447 1.0436515 -0.548889110 1.218115995 -0.80534201

13 0.6701770 0.2748662 -0.247789868 0.182578819 -0.17507435

14 -0.4272940 0.7873897 -0.669328806 0.584113643 -0.49020818

15 -0.4272940 0.6409544 -1.602736456 -0.176689180 -0.38516357

16 -1.2503973 -0.2010484 -0.037020399 -0.599357415 -0.73531227

17 -0.1529263 -1.2993131 0.565178085 -1.317893414 -0.35014870

18 0.3958092 2.1419161 -1.120977669 2.591787758 -0.80534201

19 1.7676480 1.0436515 -0.428449413 1.598517406 -0.98041636

20 0.3958092 -0.2376573 0.083419298 -0.303489651 0.21008922

21 -0.9760295 0.0552133 0.053309374 -0.134422357 0.59525279

22 -0.4272940 1.4463485 -1.030647896 1.196982583 1.40059480

23 -0.4272940 0.2016486 0.324298692 0.267112466 1.50563941

24 -1.2503973 -0.2376573 0.836167403 -0.409156709 -0.17507435

25 -1.2503973 -1.2993131 1.588915507 -1.423560473 0.17507435

26 0.9445447 0.1650398 -0.247789868 0.161445408 0.80534201

27 -0.1529263 0.2382574 -0.006910474 0.267112466 -0.42017844

28 -0.7016618 -2.8368835 2.311553687 -2.332297178 0.07002974

29 2.0420157 0.2382574 -0.910208200 0.309379290 -1.36557993

30 0.1214415 0.9704338 -1.482296759 0.816581172 -1.61068401

31 1.2189125 -0.2742661 -0.609108958 -0.514823768 -0.87537175

32 -0.9760295 0.3846927 -0.338119640 -0.007621886 -0.35014870

33 -0.9760295 0.0552133 0.294188767 -0.218956004 -0.28011896

34 0.6701770 0.6409544 0.053309374 0.816581172 0.03501487

35 0.3958092 -0.1644396 -0.247789868 -0.049888710 0.28011896

36 -0.1529263 -0.7501807 -0.187570019 -0.937492003 0.42017844

37 -0.1529263 -0.8233984 0.986717024 -0.240089415 1.82077323

38 -0.1529263 -0.9332248 0.474848312 -0.789558121 1.19050558

39 -0.4272940 -1.0796601 0.866277327 -0.641624238 1.22552045

40 0.6701770 -0.7867896 0.595288009 -0.578224003 1.75074349

41 -0.1529263 -0.3108749 0.595288009 0.309379290 1.50563941

42 0.6701770 -1.4457483 0.806057478 -1.085425885 1.96083271

43 0.6701770 0.3480839 -0.488669261 0.816581172 -0.14005948

44 -0.1529263 0.6409544 -0.639218882 0.710914113 0.35014870

45 -0.4272940 -0.4939190 0.233968919 -1.022025650 0.28011896

46 0.3958092 -1.4457483 0.986717024 -1.508094120 0.77032714

47 -0.9760295 -0.5671366 0.113529222 -0.916358591 -0.42017844

48 -0.1529263 0.4579103 -0.849988351 0.288245878 -0.17507435

49 1.7676480 -0.6403543 -1.452186835 -1.423560473 -1.78575836

50 1.7676480 -0.1278308 -0.458559337 -0.620490827 -1.22552045

51 -0.7016618 1.1534779 -0.368229565 1.154715760 0.84035688

52 0.1214415 0.3114750 -0.669328806 0.098045173 -1.01543123

53 -0.1529263 -0.4207014 0.384518540 -0.493690356 0.07002974

54 -1.2503973 0.6043456 0.565178085 0.626380466 -0.14005948

55 0.9445447 0.1284309 0.655507858 0.499579996 0.84035688

56 -1.7991328 -0.4573102 0.504958237 -1.212226355 0.42017844

57 -1.7991328 0.3480839 -0.067130323 -0.282356239 0.28011896

58 -1.5247650 -0.4573102 0.956607099 -0.345756474 0.31513383

59 -1.7991328 0.1284309 -0.127350171 -0.240089415 0.49020818

60 -0.7016618 0.2016486 0.053309374 -0.197822592 0.49020818

61 -1.2503973 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

>

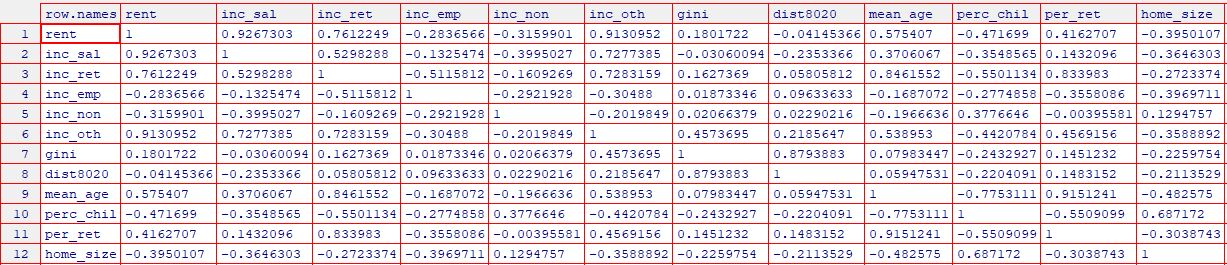
***> #Ahora vamos a calcular la matriz de covarianza de los datos Xs y la vamos a***

***> #guardar en la variable CXs.***

**> CXs <- cov(Xs)**

**> fix(CXs)**

**> CXs**



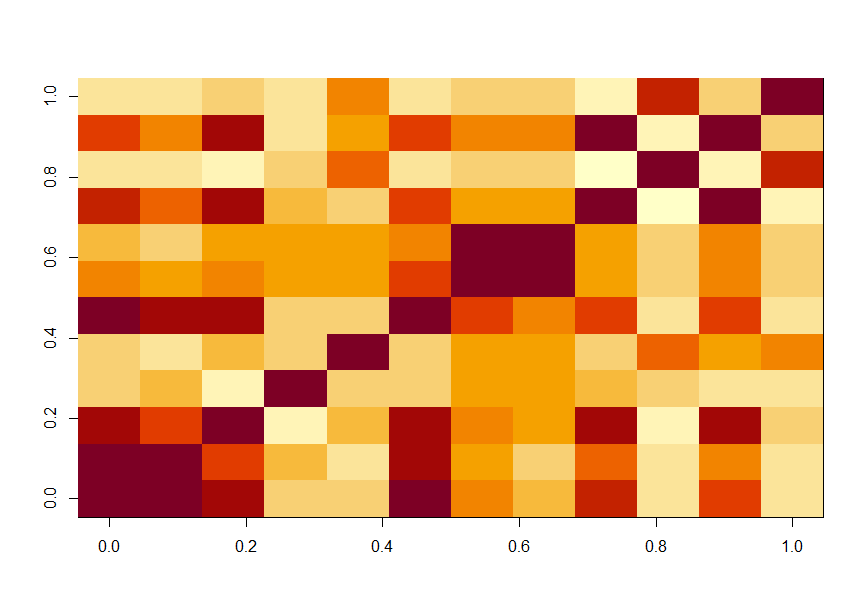
***> #Visualizamos la matriz como una imagen y la guardamos en formato .jpeg***

**> image(CXs)**

**> jpeg('CXs.jpeg')**

**> image(CXs)**

**> dev.off()**



***> #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**

***> #Visualizamos la matriz de covarianzas definida positiva como una imagen y***

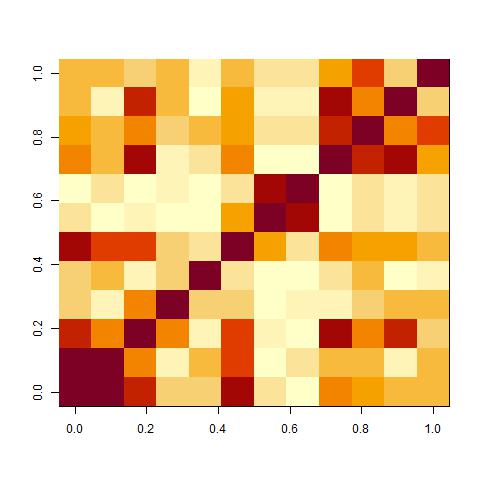
***> #la guardamos en formato .jpeg***

**> image(CXs\_ab)**

**> jpeg('CXs\_ab.jpeg')**

**> image(CXs\_ab)**

**> dev.off()**



***> #Mostramos el par de valores máximo y mínimo de los valores absolutos de la***

***> #matriz CXs\_ab.***

**> par\_max = tail(sort(CXs\_ab), 2)**

**> par\_min = head(sort(CXs\_ab), 2)**

**> par\_max**

[1] 1 1

**> par\_min**

[1] 0.00395581 0.00395581

***> #Viendo un resumen de los datos podemos observar que el valor máximo puede***

***> #corresponder a cualquier variable, ya que todas comparten como valor máximo 1.***

***> # Como valor mínimo podemos observar que las variables son inc\_non(5) y per\_ret(11)***

**> summary(CXs\_ab)**

rent inc\_sal inc\_ret inc\_emp **inc\_non** inc\_oth gini dist8020

Min. :0.04145 Min. :0.0306 Min. :0.05806 Min. :0.01873 Min. :0.003956 Min. :0.2020 Min. :0.01873 Min. :0.02290

1st Qu.:0.30791 1st Qu.:0.2123 1st Qu.:0.24494 1st Qu.:0.15967 1st Qu.:0.102832 1st Qu.:0.3454 1st Qu.:0.06753 1st Qu.:0.05912

Median :0.44398 Median :0.3676 Median :0.53997 Median :0.28792 Median :0.199324 Median :0.4571 Median :0.17145 Median :0.17983

Mean :0.52339 Mean :0.4346 Mean :0.53461 Mean :0.31991 Mean :0.260160 Mean :0.5291 Mean :0.28699 Mean :0.26597

3rd Qu.:0.79919 3rd Qu.:0.5793 3rd Qu.:0.77941 3rd Qu.:0.36610 3rd Qu.:0.331409 3rd Qu.:0.7279 3rd Qu.:0.29681 3rd Qu.:0.22414

Max. :1.00000 Max. :1.0000 Max. :1.00000 Max. :1.00000 Max. :1.000000 Max. :1.0000 Max. :1.00000 Max. :1.00000

mean\_age perc\_chil **per\_ret** home\_size

Min. :0.05948 Min. :0.2204 Min. :0.003956 Min. :0.1295

1st Qu.:0.18967 1st Qu.:0.3355 1st Qu.:0.147517 1st Qu.:0.2607

Median :0.51076 Median :0.4569 Median :0.386040 Median :0.3618

Mean :0.50073 Mean :0.4959 Mean :0.439458 Mean :0.4024

3rd Qu.:0.79302 3rd Qu.:0.5850 3rd Qu.:0.621678 3rd Qu.:0.4184

Max. :1.00000 Max. :1.0000 Max. :1.000000 Max. :1.0000

***> #Comprobamos que efectivamente los datos de maximo concuerdan con las variables***

***> # 1 y 2, y los datos de mínimo con 5 y 11***

**> CXs\_ab[1,1]**

[1] 1

**> CXs\_ab[2,2]**

[1] 1

**> CXs\_ab[11,5]**

[1] 0.00395581

**> CXs\_ab[5,11]**

[1] 0.00395581

***Pregunta 4.*** [5 %] Finalmente, calcular la descomposición en componentes principales de la matriz de covarianzas ***CXs***. Dibujar la distribución de la varianza explicada en porcentaje (eje de ordenadas) para cada componente principal (eje de abscisas) respecto la variancia total de los datos.

***Respuesta:***

>

***> #Calculamos lo componentes principales de la matriz de datos normalizada Xs***

**> comp\_prin <- prcomp(Xs, scale = TRUE, center = TRUE)**

***> #Comprobamos que la media es prácticamente cero y la varianza uno***

**> comp\_prin$center**

**rent inc\_sal inc\_ret inc\_emp inc\_non**

-2.096740e-16 1.458732e-16 -1.292227e-16 -3.150940e-16 -8.502989e-18

**inc\_oth gini dist8020 mean\_age perc\_chil**

9.577949e-17 -3.434184e-16 2.684556e-16 -4.436342e-16 4.555498e-16

**per\_ret home\_size**

1.550302e-16 3.347732e-16

**> comp\_prin$scale**

**rent inc\_sal inc\_ret inc\_emp inc\_non inc\_oth gini dist8020**

1 1 1 1 1 1 1 1

**mean\_age perc\_chil per\_ret home\_size**

1 1 1 1

**> comp\_prin$sdev^2**

[1] 5.218483e+00 2.125341e+00 1.836521e+00 1.425191e+00 7.895196e-01

[6] 2.145532e-01 1.647613e-01 1.015204e-01 5.855235e-02 5.127084e-02

[11] 1.428599e-02 2.020722e-07

***> #Ahora dibujamos la distribución de la varianza explicada en porcentaje (eje***

***> #de ordenadas) para cada componente principal (eje de abscisas) respecto a la***

***> #varianza total de los datos.***

> ***#Calculamos el % de cada valor respecto al total***

**> porc\_varianza <- abs(comp\_prin$sdev^2)\*100/sum(abs(comp\_prin$sdev^2))**

**> porc\_varianza**

[1] 4.348736e+01 1.771117e+01 1.530434e+01 1.187660e+01 6.579330e+00

[6] 1.787943e+00 1.373010e+00 8.460034e-01 4.879362e-01 4.272570e-01

[11] 1.190499e-01 1.683935e-06

>

> ***#Importamos paquete ggplot2 con el que vamos a generar la gráfica.***

**> library(ggplot2)**

**> library(crayon)**

**Attaching package: ‘crayon’**

**The following object is masked from ‘package:ggplot2’:**

**%+%**

**> ggplot(data = data.frame(porc\_varianza, pc = 1:12),**

**+ aes(x = pc, y = porc\_varianza)) + geom\_col(width = 0.5) +**

**+ geom\_text(aes(label = round(porc\_varianza, digits = 2)), vjust = -1,**

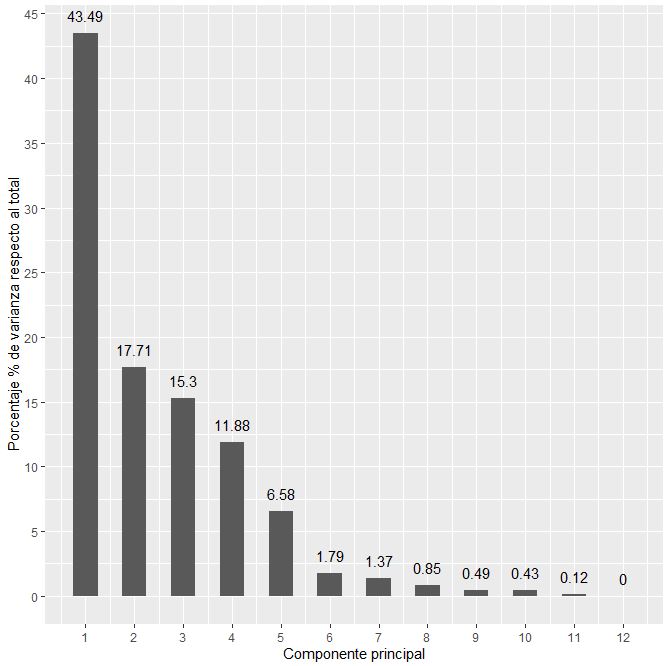
**+ colour = "black") + ylim(c(0, 100)) +**

**+ scale\_x\_continuous(breaks = seq(1, 12, by = 1)) +**

**+ scale\_y\_continuous(breaks = seq(0, 60, by = 5)) +**

**+ labs(x = "Componente principal",**

**+ y = "Porcentaje % de varianza respecto al total")**



***Pregunta 5.*** [20 %] Como habéis visto, la mayor parte de la varianza queda concentrada en unas pocas componentes principales. Por esto, podemos reducir la dimensión del subespacio, proyectar nuestros datos allí y utilizar estas representaciones para análisis posteriores. Un buen criterio para el diseño del nuevo subespacio es restringir el porcentaje total de varianza explicada por el subespacio a un cierto umbral. En esta práctica, os quedareis con las ***L*** primeras componentes principales que expliquen, al menos, un 75% de la varianza inicial. Calcular el valor mínimo de ***L***, es decir, el mínimo número de componentes principales necesarias para explicar un 75% de la varianza de nuestros datos.

***Respuesta:***

* ***L*** = 3
* Copiamos código R usado para conseguir la respuesta:

> #Calculamos la variabilidad acumuladada de las variables originales y de las

> #componentes principales

**> library(data.table)**

**> variabilidad\_org = 100/12**

**> variabilidad\_org**

[1] 8.333333

**> com\_principales <- round(porc\_varianza, digits = 2)**

**> variabilidad\_cp = data.frame(variables = 1:12, com\_principales,**

**+ variabilidad\_org )**

**> dt <- as.data.table(variabilidad\_cp)**

**> dt[, acumulado\_com.prin:= cumsum(com\_principales)]**

**> dt[,acumulado\_original := cumsum(variabilidad\_org)]**

**> dt**

variables com\_principales variabilidad\_org acumulado\_com.prin acumulado\_original

1: 1 43.49 8.333333 43.49 8.333333

2: 2 17.71 8.333333 61.20 16.666667

3: 3 15.30 8.333333 76.50 25.000000

4: 4 11.88 8.333333 88.38 33.333333

5: 5 6.58 8.333333 94.96 41.666667

6: 6 1.79 8.333333 96.75 50.000000

7: 7 1.37 8.333333 98.12 58.333333

8: 8 0.85 8.333333 98.97 66.666667

9: 9 0.49 8.333333 99.46 75.000000

10: 10 0.43 8.333333 99.89 83.333333

11: 11 0.12 8.333333 100.01 91.666667

12: 12 0.00 8.333333 100.01 100.000000

> #Podemos observar en la columna acumulado\_com.prin que necesitamos como

> #mínimo de 3 componentes principales para explicar un 75% de la varianza

***Pregunta 6.*** [10 %] Considerar la componente principal C e indicar que variables contribuyen en mayor y menor peso (en valor absoluto).

***Respuesta:***

* ***C*** = 1
* Variable que contribuye en mayor peso = 3
* Variable que contribuye en menor peso = 8
* Copiamos código R usado para conseguir la respuesta:

>#Obtenemos la carga de cada variable en la componente principal C = 1 y las

>#ordenamos en valores absolutos para detectar el valor máximo y el valor mínimo.

**> loadings\_cpC1 <- comp\_prin$rotation[,1]**

**> loadings\_cpC1\_abs <- abs(loadings\_cpC1)**

**> loadings\_cpC1\_ordmax <- sort(loadings\_cpC1\_abs, decreasing = TRUE)**

**> loadings\_cpC1\_ordmin <- sort(loadings\_cpC1\_abs, decreasing = FALSE)**

**> loadings\_cpC1\_ordmax[0:1]**

inc\_ret

0.393225

**> loadings\_cpC1\_ordmin[0:1]**

dist8020

0.07551688

#Comprobamos que sus posiciones son 3 y 8 respectivamente

**> loadings\_cpC1[3:3]**

inc\_ret

-0.393225

**> loadings\_cpC1[8:8]**

dist8020

-0.07551688

***Pregunta 7.*** [10 %] Calcular las nuevas variables proyectadas a las componentes principales. Para la componente principal C, anotar las secciones censales (relacionarlo con la variable id ) con el valor máximo y mínimo.

***Respuesta:***

* ***C*** = 1
* Número de las dos regiones censales = {5 , 44 }
* Copiamos código R usado para conseguir la respuesta:

# Cálculo de las nuevas variables proyectadas

**> nuevas\_vars <- predict(comp\_prin, newdata = Xs)**

**> nuevas\_vars**

PC1 PC2 PC3 PC4 PC5 PC6 PC7

1 -0.50794175 4.160920259 -1.358941167 1.377985127 -0.743106650 0.7389278499 0.26811024

2 -1.16093410 1.176711951 2.718741157 0.279414147 1.659629993 -0.8997296842 -0.23911466

3 -0.70326104 -0.008370836 -0.623620313 -0.120038497 -0.288280861 -0.4870146888 0.25049441

4 0.66376338 1.561242089 -0.522259943 -0.372580365 -0.171668850 -0.4806855247 0.15148039

**5 7.53212491** -1.418681812 5.844886318 2.095023111 -2.229412686 -0.0219708550 0.33604441

6 2.85603905 -0.232678915 1.898477497 -1.460840712 -0.867420117 -0.0490803707 -0.47406645

7 -3.44837180 -0.793740884 0.701570102 0.853413972 -0.694867623 0.1437992747 -0.85390217

8 -1.65193465 1.106928360 0.634993608 0.976624710 -1.734932206 0.0579745415 0.25884832

9 -3.00719695 -0.435952235 1.181810561 -0.401342367 -0.388761024 -0.0369140157 -0.41558901

10 -2.92144558 0.721523613 0.864824177 -0.461269197 -0.086813947 -0.1827376379 -0.24715240

11 -2.40611561 -0.552045154 0.290695782 -0.636536564 -1.620055092 -0.0005384315 -0.33873119

12 -2.41155289 0.617334033 0.870641832 -0.361522916 0.828362048 -0.9548757813 -0.04624334

13 -1.83097026 0.275114641 0.882997097 1.244024296 -0.392906354 0.1181731956 0.20669120

14 -2.25081180 -0.272663553 -0.005282770 0.183468291 -0.297202653 0.4244824891 -0.13968576

15 -0.73101713 0.051583193 -1.122406227 -1.045611009 -0.194308554 0.4587193303 1.00883543

16 0.36727677 -1.062539284 -1.868893976 -0.210968717 -0.252516486 -0.6641057260 0.21596271

17 1.79068248 0.433502859 -1.720314740 0.888885825 0.001468144 -0.4836690379 -0.10196197

18 -3.73303788 -0.252856886 1.469450114 -1.879798360 0.130590874 -0.3663400721 -0.13079873

19 -2.77633499 1.904522730 1.155564594 -0.249036129 0.781751335 -0.5739087256 -0.76059482

20 1.36494487 0.855527711 0.177098411 -0.448515334 -0.548006405 0.5598166915 -0.05513769

21 -1.58318379 -1.891240583 -0.776358592 1.486868392 0.340187356 0.3207802734 0.02346615

22 -4.88036789 -3.110858674 1.350679044 1.022277179 1.858903435 0.1707335990 0.94569094

23 -0.67123817 -0.943954176 -0.371341678 1.055805592 1.209474840 1.0015686521 -0.59543610

24 0.22750479 -1.658635139 -2.032573575 0.486574678 0.266937814 -0.5525423189 -0.69510498

25 1.46202412 -1.918865766 -1.641496323 1.826394337 -0.224572745 -0.6704005039 -0.41786845

26 -2.96483977 0.140045063 1.462419682 2.996109375 0.308887635 0.9424639041 0.01826190

27 -0.47878183 -0.369162542 0.035019802 -0.245828459 -0.235084202 -0.4973974281 0.12658472

28 3.95185691 -0.634462876 -1.043137224 2.249050127 -0.977221690 -0.8414931022 -0.33917673

29 -2.02109901 2.663297448 0.164933992 0.698632233 -0.126895871 -0.2396369692 -0.16020433

30 -3.23961250 0.971675702 -0.627586193 -0.157155661 -0.730065366 0.3762966735 -0.20618860

31 -0.63223387 1.245865280 0.040672816 0.907578953 -0.974034315 -0.2098520340 0.58758352

32 -1.38682197 -1.903982656 -0.580892859 0.334793617 -0.404088325 -0.4196169455 0.49287760

33 -1.33446967 -2.279825003 -0.342008989 0.841062625 -0.378342989 -0.7239472662 0.30371264

34 -0.53126893 -0.051946395 1.722240959 -0.819890083 -0.208103451 -0.4316601392 0.31131555

35 1.19099293 0.935144859 0.344764697 -0.642497904 1.139335222 -0.4690341696 0.51418931

36 1.95395328 0.553139396 -1.094632848 -0.133476254 0.770276040 0.0519168927 0.44011952

37 3.02290277 -0.212658036 0.836872233 -0.482024344 1.566080140 0.1441154435 -0.10696977

38 2.38589042 0.100517509 0.046984408 0.107994163 0.365334792 0.5810676069 0.19084341

39 2.77496509 0.230245959 -0.160954095 0.007208606 1.173979284 0.2879278600 -0.30123734

40 2.53309427 0.717192172 0.223078520 0.008091281 1.754173502 0.4653467621 -0.11105555

41 2.35791401 -0.403886734 1.115558697 -1.324670774 0.816518686 0.3736704289 -0.07437692

42 3.24977327 0.520598118 0.682296342 0.693252505 1.883465964 0.0046686206 0.45269993

43 0.14844835 1.076913444 0.446207524 -1.408803103 0.988219616 -0.4609375752 0.08012771

**44 -0.02298756** 0.336625432 -0.005377663 -1.499240714 1.051078028 0.2284079683 0.10350660

45 2.01525020 -0.048104574 -0.922663438 -0.220875423 0.313043001 -0.1386923455 0.30697305

46 3.13031957 0.656955584 -1.000795419 0.718049973 0.843956642 -0.1627779224 -0.18113491

47 0.18877462 -1.001959781 -1.228200990 0.809786569 -0.911003694 -0.1046766557 0.30145481

48 -1.10413083 -0.278702650 -0.284343326 -0.465956112 -0.435019741 0.3498028396 0.50535751

49 -0.33350198 4.048669925 -2.542005313 1.369831763 -0.084208625 0.2073324328 0.07299176

50 -0.12202331 3.267095911 0.042918061 0.905553260 -0.610201895 -0.0791005335 0.08033796

51 -3.45157212 -2.189301067 1.133516676 0.837474387 0.411285729 0.6556821617 0.01576882

52 0.55853126 1.271048166 -0.519566610 -1.353987937 -0.567473432 -0.1989154026 0.21119984

53 1.84920961 0.216173754 -0.755716552 -0.773764234 0.066353202 -0.0002145597 -0.30442385

54 -0.15857020 -1.174274029 -0.599231247 -0.903968582 0.148464617 -0.0726752709 -0.94526472

55 1.68281255 1.304433614 1.666902707 -0.975079101 -0.152538106 0.7644755525 -0.61078482

56 0.70805968 -2.057852596 -2.707888045 0.861912207 -0.060862002 0.4055729254 -0.26630616

57 0.77465011 -2.079506926 -0.900795004 -1.272611006 -0.350789145 0.1373955709 0.43980675

58 2.18067352 -1.361775046 -0.945819452 -0.936830785 -0.957188613 0.4867769627 -0.67983350

59 1.40982317 -1.700365175 -1.370563144 -1.577320271 0.203376907 0.3333050024 0.14550188

60 0.95281684 -0.830478916 -0.075503493 -0.873745771 -0.337210687 0.4383045848 0.21545445

61 -0.82744297 0.010780128 -0.255646203 -4.407354614 -1.645970437 0.2456356030 0.21605150

PC8 PC9 PC10 PC11 PC12

1 0.316571377 0.05233203 0.104479128 -0.0022552549 1.571576e-04

2 -0.638730802 -0.46522267 -0.341499216 -0.0282859459 -5.344653e-04

3 0.340913758 0.33230971 -0.303947486 0.0917640209 -5.879721e-05

4 0.124763511 -0.10919148 -0.124803516 -0.0006297962 1.160190e-04

5 0.467811617 0.19388971 0.305492603 -0.0679023641 -1.619465e-03

6 -0.159311208 -0.49051711 -0.022957126 -0.0111954757 9.974688e-04

7 0.426330906 -0.30768299 -0.367395819 -0.0152434712 -1.208613e-04

8 -0.732790750 0.05317348 -0.161830900 -0.0431480228 2.612461e-04

9 0.599849027 0.06478591 0.066408706 -0.0885876705 3.938604e-04

10 -0.007193477 -0.29454916 -0.140275984 0.0487716264 -3.762498e-04

11 -0.383544074 -0.05202353 0.235074024 0.2756136765 -1.042364e-04

12 0.226977125 0.02889947 0.053375773 -0.0972329028 -2.295225e-05

13 -0.124391576 0.20406990 -0.165445136 -0.0287829626 5.935276e-04

14 0.573317420 0.01869240 0.061112657 -0.0955890657 2.864816e-04

15 -0.145832896 -0.02930164 0.123485708 -0.0886265555 -2.661810e-04

16 0.123360660 0.15569162 -0.065560658 -0.0326834838 -2.448013e-04

17 -0.129769817 0.22481351 0.180824318 0.1169239880 8.939872e-05

18 -0.234935062 0.26040959 0.010414538 0.0893126198 -3.188739e-04

19 -0.041622044 0.30310279 0.513039339 -0.0554725426 -6.537855e-05

20 -0.348151954 -0.13773034 0.008618919 -0.0369510874 3.558797e-04

21 -0.163301829 0.24496521 -0.133332557 0.1135206747 -3.022725e-04

22 -0.028373983 -0.10876730 0.637521986 -0.0549815336 2.993143e-04

23 -0.416118088 0.35854228 -0.369413905 0.1079080676 -9.171161e-04

24 -0.245306597 0.06256866 -0.047995880 -0.0275953210 -8.214082e-04

25 -0.097647223 0.00658328 -0.151865830 0.0101604968 4.406103e-05

26 0.324017582 -0.17021649 -0.391617784 0.0901763095 2.447617e-04

27 -0.174697078 0.30509248 -0.099321921 -0.0002994829 2.382381e-04

28 0.157438606 0.17449732 0.178895249 0.2454278895 1.199507e-03

29 -0.075850619 -0.24630258 0.426611907 0.0616358203 -2.308706e-04

30 0.562472764 -0.04939042 0.480329814 0.0370018458 5.397045e-05

31 -0.372021328 -0.43285774 -0.096218635 0.1311230251 -6.367735e-05

32 0.181752963 -0.49968297 -0.341308879 0.0486458767 -1.212569e-04

33 0.036217901 -0.17762844 0.173138909 -0.0741739499 3.335915e-04

34 -0.574836570 0.30745647 -0.010432309 -0.1171582548 4.558822e-04

35 0.430776460 0.19380745 -0.346352529 0.0090685961 2.836349e-05

36 0.164877654 -0.02419910 -0.089934431 0.0755228205 -3.711807e-04

37 0.342449897 -0.06021424 -0.159185560 0.1020413953 4.097408e-04

38 -0.124207794 0.16502463 0.010492709 0.1015444173 3.003465e-04

39 0.475395110 0.36824858 0.073533795 0.0721964234 4.493188e-04

40 -0.302672474 -0.11255004 0.183290780 0.0066737181 -1.637943e-04

41 -0.021404937 -0.21974411 0.029003795 0.1663277633 3.769272e-04

42 0.124340983 -0.16656441 -0.111461373 0.1409567783 3.050650e-04

43 0.065424393 0.16237847 0.111847932 0.1448615167 -1.458806e-04

44 0.067541300 0.44042193 0.112601532 -0.0342018313 -2.341120e-04

45 0.138794793 -0.19562218 -0.197529620 -0.2307154062 1.459368e-04

46 -0.257581890 -0.31531431 0.190081988 -0.0331638057 -1.076899e-04

47 -0.004119639 0.14350069 0.012658523 0.0346826603 3.748274e-04

48 -0.619072503 0.38361171 0.219717713 0.1146717264 -1.726807e-04

49 0.204568208 -0.34058054 0.194869307 -0.0259086731 -8.341841e-04

50 0.011636245 0.39661053 -0.328293086 -0.3116924321 2.318727e-04

51 0.214918791 -0.02215440 -0.060346315 0.0111730651 2.132221e-04

52 0.109563729 0.14487230 -0.254061010 -0.0653914073 -2.575271e-04

53 -0.081735680 -0.39052689 0.358801766 -0.0928449174 -1.260925e-04

54 0.385525805 0.10202168 -0.003967989 -0.2203050005 -2.646952e-04

55 -0.489334269 0.12083869 -0.141083522 -0.1346276346 5.339248e-04

56 -0.421229976 0.03897256 -0.003503670 -0.1200867684 -5.603405e-04

57 0.009493622 -0.02444772 0.016762984 -0.2547576679 2.387585e-04

58 -0.184185898 -0.07878713 0.144655242 0.0271711917 3.216706e-04

59 0.242619095 -0.32654287 0.082752906 -0.0681963827 -3.954628e-04

60 -0.212072882 -0.11569446 -0.021358650 -0.2069686398 2.768668e-04

61 0.362323616 -0.04817782 -0.247593254 0.2907777006 -5.047028e-04

#También podríamos calcularlas así

**comp\_prin$x**

#Calculamos valores máximos y mínimos en C=1

**> max <- max(abs(nuevas\_vars[,1]))**

**> min <- min(abs(nuevas\_vars[,1]))**

**> max**

[1] 7.532125

**> min**

[1] 0.02298756

**> ord <- sort(abs(nuevas\_vars[,1]), decreasing = FALSE)**

**> ord**

**44** 50 43 54 47 24 49 16 27 1 34 52 31

**0.02298756** 0.12202331 0.14844835 0.15857020 0.18877462 0.22750479 0.33350198 0.36727677 0.47878183 0.50794175 0.53126893 0.55853126 0.63223387

4 23 3 56 15 57 61 60 48 2 35 33 20

0.66376338 0.67123817 0.70326104 0.70805968 0.73101713 0.77465011 0.82744297 0.95281684 1.10413083 1.16093410 1.19099293 1.33446967 1.36494487

32 59 25 21 8 55 17 13 53 36 45 29 58

1.38682197 1.40982317 1.46202412 1.58318379 1.65193465 1.68281255 1.79068248 1.83097026 1.84920961 1.95395328 2.01525020 2.02109901 2.18067352

14 41 38 11 12 40 39 19 6 10 26 9 37

2.25081180 2.35791401 2.38589042 2.40611561 2.41155289 2.53309427 2.77496509 2.77633499 2.85603905 2.92144558 2.96483977 3.00719695 3.02290277

46 30 42 7 51 18 28 22 **5**

3.13031957 3.23961250 3.24977327 3.44837180 3.45157212 3.73303788 3.95185691 4.88036789 **7.53212491**

> #comprobamos que el id 44 coincide con min y que el id 5 coincide con max

**> nuevas\_vars[44]**

[1] -0.02298756

**> nuevas\_vars[5]**

[1] 7.532125

***Pregunta 8.*** [20 %] Cuando reducimos la dimensión del subespacio generado por los datos iniciales a ***L***, se produce una pérdida de información. Una manera de medir el error cometido en esta aproximación es calculando el error residual, tal y como se indica en la Sección 2.5.1 de los apuntes del módulo. Considerando el valor de ***L*** calculado en el apartado 5, calcular la desviación típica del error residual cuando se consideran solo las ***L*** primeras componentes principales.

***Respuesta:***

* ***L*** = 3
* Desviación típica = 4.801111
* Copiamos código R usado para conseguir la respuesta:

> #Tomamos solo las tres primeras componentes principales

**> L <- comp\_prin$x[, 1:3]**

**> L**

PC1 PC2 PC3

1 -0.50794175 4.160920259 -1.358941167

2 -1.16093410 1.176711951 2.718741157

3 -0.70326104 -0.008370836 -0.623620313

4 0.66376338 1.561242089 -0.522259943

5 7.53212491 -1.418681812 5.844886318

6 2.85603905 -0.232678915 1.898477497

7 -3.44837180 -0.793740884 0.701570102

8 -1.65193465 1.106928360 0.634993608

9 -3.00719695 -0.435952235 1.181810561

10 -2.92144558 0.721523613 0.864824177

11 -2.40611561 -0.552045154 0.290695782

12 -2.41155289 0.617334033 0.870641832

13 -1.83097026 0.275114641 0.882997097

14 -2.25081180 -0.272663553 -0.005282770

15 -0.73101713 0.051583193 -1.122406227

16 0.36727677 -1.062539284 -1.868893976

17 1.79068248 0.433502859 -1.720314740

18 -3.73303788 -0.252856886 1.469450114

19 -2.77633499 1.904522730 1.155564594

20 1.36494487 0.855527711 0.177098411

21 -1.58318379 -1.891240583 -0.776358592

22 -4.88036789 -3.110858674 1.350679044

23 -0.67123817 -0.943954176 -0.371341678

24 0.22750479 -1.658635139 -2.032573575

25 1.46202412 -1.918865766 -1.641496323

26 -2.96483977 0.140045063 1.462419682

27 -0.47878183 -0.369162542 0.035019802

28 3.95185691 -0.634462876 -1.043137224

29 -2.02109901 2.663297448 0.164933992

30 -3.23961250 0.971675702 -0.627586193

31 -0.63223387 1.245865280 0.040672816

32 -1.38682197 -1.903982656 -0.580892859

33 -1.33446967 -2.279825003 -0.342008989

34 -0.53126893 -0.051946395 1.722240959

35 1.19099293 0.935144859 0.344764697

36 1.95395328 0.553139396 -1.094632848

37 3.02290277 -0.212658036 0.836872233

38 2.38589042 0.100517509 0.046984408

39 2.77496509 0.230245959 -0.160954095

40 2.53309427 0.717192172 0.223078520

41 2.35791401 -0.403886734 1.115558697

42 3.24977327 0.520598118 0.682296342

43 0.14844835 1.076913444 0.446207524

44 -0.02298756 0.336625432 -0.005377663

45 2.01525020 -0.048104574 -0.922663438

46 3.13031957 0.656955584 -1.000795419

47 0.18877462 -1.001959781 -1.228200990

48 -1.10413083 -0.278702650 -0.284343326

49 -0.33350198 4.048669925 -2.542005313

50 -0.12202331 3.267095911 0.042918061

51 -3.45157212 -2.189301067 1.133516676

52 0.55853126 1.271048166 -0.519566610

53 1.84920961 0.216173754 -0.755716552

54 -0.15857020 -1.174274029 -0.599231247

55 1.68281255 1.304433614 1.666902707

56 0.70805968 -2.057852596 -2.707888045

57 0.77465011 -2.079506926 -0.900795004

58 2.18067352 -1.361775046 -0.945819452

59 1.40982317 -1.700365175 -1.370563144

60 0.95281684 -0.830478916 -0.075503493

61 -0.82744297 0.010780128 -0.255646203

#Calculamos los datos originales con pérdida de información

**dtorg\_lossinf <- nuevas\_vars[,1:3]%\*%t(L)**

**org\_lossinf**

1 2 3 4 5 6

1 19.41798332 1.79128221 1.16984858 6.8687712 -17.6717592 -4.9987791

2 1.79128221 10.12397247 -0.88887255 -0.3533429 5.4770525 1.5719997

3 1.16984858 -0.88887255 0.88354846 -0.1541759 -8.9301643 -3.1905224

4 6.86877124 -0.35334291 -0.15417591 3.1508141 -0.2679071 0.5409673

5 -17.67175924 5.47705252 -8.93016429 -0.2679071 92.9082598 32.9385254

6 -4.99877911 1.57199973 -3.19052241 0.5409673 32.9385254 11.8153153

7 -2.50451301 4.97671553 1.99423645 -3.8945265 -20.7469039 -8.3320827

8 4.58200826 4.94670635 0.75648045 0.3000577 -10.3014918 -3.7700277

9 -1.89249263 6.19120428 1.38149267 -3.2939065 -15.1245572 -6.2435943

10 3.31088122 6.59186432 1.50917718 -1.2643386 -17.9735064 -6.8697967

11 -1.46988776 2.93407010 1.51546465 -2.6107863 -15.6409031 -6.1916315

12 2.61045505 5.89312809 1.14783366 -1.0915940 -13.9511156 -5.3782359

13 0.87481521 4.85000703 0.73469218 -1.2469645 -9.0203792 -3.6169858

14 0.01592895 2.27783521 1.58848511 -1.9169413 -16.5974500 -6.3749925

15 2.11123170 -2.14217074 1.21361939 0.1814993 -12.1396292 -4.2306787

16 -2.06797947 -6.75772577 0.91608315 -0.4390477 -6.6496932 -2.2518659

7 8 9 10 11 12

1 -2.5045130 4.58200826 -1.89249263 3.3108812 -1.4698878 2.6104551

2 4.9767155 4.94670635 6.19120428 6.5918643 2.9340701 5.8931281

3 1.9942365 0.75648045 1.38149267 1.5091772 1.5154646 1.1478337

4 -3.8945265 0.30005770 -3.29390649 -1.2643386 -2.6107863 -1.0915940

5 -20.7469039 -10.30149179 -15.12455717 -17.9735064 -15.6409031 -13.9511156

6 -8.3320827 -3.77002768 -6.24359427 -6.8697967 -6.1916315 -5.3782359

7 13.0134933 5.26336310 11.54508924 10.1082626 8.9393055 8.4367440

8 5.2633631 4.35739536 5.23556710 6.1738700 3.5482613 5.2199243

9 11.5450892 5.23556710 10.62996405 9.4928707 7.8198762 8.0118200

10 10.1082626 6.17386995 9.49287075 9.8033614 6.8824229 8.2435937

11 8.9393055 3.54826126 7.81987617 6.8824229 6.1786502 5.7147707

12 8.4367440 5.21992432 8.01182004 8.2435937 5.7147707 6.9547058

13 6.7149808 3.88987292 6.42968662 6.3112189 4.5103339 5.3540934

14 7.9743539 3.41302045 6.88125942 6.3743223 5.5647003 5.2550278

15 1.6924285 0.55197064 0.84935313 1.2021612 1.4041567 0.8175167

16 -1.7342861 -2.96960781 -2.84993584 -3.4558910 -0.8404203 -3.1687863

13 14 15 16 17 18

1 0.8748152 0.01592895 2.11123170 -2.0679795 3.23201496 -1.152847797

2 4.8500070 2.27783521 -2.14217074 -6.7577258 -6.24584684 8.031325748

3 0.7346922 1.58848511 1.21361939 0.9160831 -0.19012279 1.711037791

4 -1.2469645 -1.91694127 0.18149927 -0.4390477 2.76384383 -3.640059572

5 -9.0203792 -16.59744996 -12.13962924 -6.6496932 2.81759739 -19.170215305

6 -3.6169858 -6.37499254 -4.23067874 -2.2518659 1.74741329 -7.813149531

7 6.7149808 7.97435391 1.69242851 -1.7342861 -7.72594930 14.104527702

8 3.8898729 3.41302045 0.55197064 -2.9696078 -3.57062269 6.819931599

9 6.4296866 6.88125942 0.84935313 -2.8499358 -7.60700756 13.072825333

10 6.3112189 6.37432231 1.20216120 -3.4558910 -6.40636864 11.994240794

11 4.5103339 5.56470030 1.40415671 -0.8404203 -5.04799045 9.548872077

12 5.3540934 5.25502779 0.81751671 -3.1687863 -5.54848741 10.125685861

13 4.2078240 4.04149104 0.36158047 -2.6150209 -4.67845629 8.063036885

14 4.0414910 5.14052706 1.63724653 -0.5270822 -4.13960165 8.463547796

15 0.3615805 1.63724653 1.79684260 1.7743635 0.64423388 1.066551502

16 -2.6150209 -0.52708221 1.77436346 4.7566466 3.41214811 -3.848634178

19 20 21 22 23 24

1 7.7644394 2.62580378 -6.01011046 -12.3005856 -3.0821367 -4.25485983

2 8.6058976 -0.09641662 -2.49819141 5.6773476 -1.3410808 -7.74189529

3 1.2159122 -1.07751620 1.61337574 2.6159022 0.7115336 1.12144308

4 0.5270865 2.14919488 -3.59808282 -8.8016185 -1.7253474 -1.37698988

5 -16.8594700 10.10233375 -13.77939718 -24.4516564 -5.8871290 -7.81349149

6 -6.1786501 4.03548994 -5.55548224 -10.6504563 -2.4024280 -2.82310321

7 8.8728473 -5.26165778 6.41589135 20.2461348 2.8034116 -0.89398743

8 7.4282703 -1.19533548 0.02886558 5.4762237 -0.1718477 -3.50248455

9 8.8843636 -4.26833050 4.66792725 17.6286600 1.9912088 -2.36318293

10 10.4844299 -3.21716973 2.58919687 13.1812711 0.9587553 -3.61920605

11 5.9647182 -3.70502332 4.62768926 13.8527005 2.0282345 -0.22262190

12 8.8770882 -2.60930109 1.97447400 11.0247840 0.7126857 -3.34221533

13 6.6277091 -2.10742788 1.69293208 9.2726113 0.6414279 -2.66762588

14 5.7236090 -3.30644082 4.08322246 11.8258721 1.7701744 -0.04908349

15 0.8307769 -1.15244359 1.93116796 1.8911539 0.8587906 2.02950564

16 -5.2029413 -0.73869742 2.87898269 -1.0113121 1.4504564 5.64458673

25 26 27 28 29 30

1 -6.49617360 0.1013399 -1.34045247 -3.2297004 11.88423328 6.5414523

2 -8.41806954 7.5827169 0.21664596 -8.1704456 5.92870952 3.1981146

3 0.01154836 1.1718894 0.31795975 -2.1233545 1.29620998 2.6615350

4 -1.16808816 -2.5130710 -0.91243938 2.1773365 2.73038215 -0.3055520

5 4.14004885 -13.9825461 -2.87783363 24.5689623 -18.03752142 -29.4478346

6 1.50573375 -5.7239128 -1.21503897 9.4539112 -6.07890739 -10.6700065

7 -4.67014529 11.1387003 1.96860603 -13.8557067 4.97124553 9.9598340

8 -5.58155491 5.9813686 0.40451716 -7.8928998 6.39153499 6.0286903

9 -5.49999835 10.5831072 1.64211528 -12.8402071 5.11169304 8.5768606

10 -7.07533657 10.0273998 1.16266155 -12.9050451 7.96880167 9.6226869

11 -2.93567456 7.4815553 1.36597894 -9.4616080 3.44068303 7.0760367

12 -6.13948499 8.5095662 0.95720082 -10.8299864 6.66172974 7.8659426

13 -4.65426723 6.7583742 0.80599566 -8.3313696 4.57892053 5.6447995

14 -2.75886473 6.6273855 1.17811997 -8.7163806 3.82205804 7.0301329

15 0.67467980 0.5331437 0.29164869 -1.7507790 1.42971644 3.1227410

16 5.64362034 -3.9708275 0.15095596 4.0750798 -3.88040502 -1.0493860

31 32 33 34 35 36

1 5.4498121 -6.4284960 -8.3435671 -2.286715279 2.81759322 2.7966161

2 2.3105853 -2.2097276 -2.0632999 5.237969644 0.65505780 -4.5935486

3 0.4088321 1.3534924 1.1708483 -0.699968873 -1.06040914 -0.6961342

4 1.5042018 -3.5897224 -4.2665132 -1.333195219 2.07046821 2.7322300

5 -6.2918229 -11.1398234 -8.8160497 6.138414105 9.65914499 7.5346868

6 -2.0183548 -4.6206131 -3.9301267 1.764397734 3.83846185 3.3737272

7 1.2198181 5.8860096 6.1713946 3.081517531 -4.60737255 -7.9449684

8 2.4493196 -0.1854964 -0.5363198 1.913732611 -0.71338075 -3.3106023

9 1.4061816 4.3139870 4.6027181 3.655638999 -3.58178225 -7.4107134

10 2.7811329 2.1753863 1.9578553 3.004028326 -2.50653109 -6.2559300

11 0.8452773 4.2190753 4.3700341 1.807619393 -3.28168723 -5.3250006

12 2.3291919 1.6632516 1.5129633 2.748569860 -1.99467913 -5.3236231

13 1.5362712 1.5024896 1.5141681 2.479180156 -1.61897437 -4.3920112

14 1.0831225 3.6436906 3.6270720 1.200852053 -2.93750217 -4.5430194

15 0.4807881 1.5675749 1.2417926 -1.547366854 -1.20936452 -0.1712179

16 -1.6319988 2.5993360 2.5714625 -3.358613402 -1.20053278 2.1756620

37 38 39 40 41 42

1 -3.5575718 -0.85749706 -0.23275840 1.3943645 -4.3942021 -0.41172886

2 -1.4843892 -2.52384294 -3.38821094 -1.4903342 -0.1797258 -1.30519143

3 -2.6460001 -1.70804562 -1.85307794 -1.9265463 -2.3505333 -2.71529064

4 1.2374166 1.71606077 2.28544975 2.6845808 0.3519204 2.61352412

5 27.9619984 18.10284080 19.63397953 19.3659836 24.8534034 27.72707970

6 10.2717925 6.88000673 7.56626755 7.4912502 8.9461335 10.45567142

7 -9.6681728 -8.27425924 -9.86478758 -9.1478104 -7.0277401 -11.14096783

8 -4.6976265 -3.80023457 -4.43140002 -3.2489724 -3.6338209 -4.35889444

9 -8.0087308 -7.16313655 -8.63546004 -7.6665384 -5.5962574 -9.19331917

10 -8.2609364 -6.85709000 -8.07997859 -6.6899023 -6.2151691 -8.52834556

11 -6.9127815 -5.78256020 -6.85078165 -6.4259921 -5.1261618 -7.90838319

12 -6.6925550 -5.65074145 -6.68996977 -5.4717222 -4.9643153 -6.92158144

13 -4.8543947 -4.29935345 -5.15965650 -4.2437325 -3.4433505 -5.20454835

14 -6.7504221 -5.39784596 -6.30785355 -5.8982491 -5.2029887 -7.46018057

15 -3.1600739 -1.79167733 -1.83601430 -2.0651250 -2.9966193 -3.11459947

16 -0.2278260 0.68166944 1.07534097 -0.2486083 -0.7897084 -0.63472925

43 44 45 46 47 48

1 3.79917808 1.419655847 0.03005632 2.5035419 -2.5959084 -0.21241951

2 2.30800093 0.408177733 -4.90466096 -5.5819509 -4.7373335 0.18081448

3 -0.39167668 0.016702037 -0.84145262 -1.5828147 0.6415605 0.95614744

4 1.54682086 0.513104032 1.74441654 3.6261335 -0.7975599 -1.01950278

5 2.19836624 -0.682141359 9.85449841 16.7964116 -4.3353590 -9.58301533

6 1.02051417 -0.154188373 4.01517041 6.8874676 -1.5594293 -3.62841193

7 -1.05364947 -0.191696533 -7.55846248 -12.0180864 -0.7173377 3.82918452

8 1.23017799 0.407179397 -3.96819533 -5.0793793 -2.2008408 1.33489191

9 -0.38856348 -0.083979878 -7.12969635 -10.8826393 -1.5823768 3.10579995

10 0.72922576 0.305389361 -6.72009402 -9.5365614 -2.3366103 2.77866059

11 -0.82197809 -0.132084983 -5.09058342 -8.1855069 -0.2581194 2.72786545

12 0.69531121 0.258564026 -5.69288841 -8.0147045 -2.1431070 2.24305605

13 0.41847009 0.129951657 -4.51780658 -6.4344834 -1.7057924 1.69388119

14 -0.63012155 -0.040016413 -4.51795834 -7.2196011 -0.1452099 2.56268486

15 -0.55379375 0.040204435 -0.44006061 -1.1311303 1.1888587 1.11191089

16 -1.92365576 -0.356070258 2.51562772 2.3220331 3.4293316 0.42201844

49 50 51 52 53 54

1 20.47002796 13.59778318 -8.8966920 5.71108917 0.98716532 -3.991196162

2 -1.75976236 4.10277493 4.5126095 -0.56532755 -3.94703394 -2.826847386

3 1.78589435 0.03170134 1.7387985 -0.07942072 -0.83100644 0.495038677

4 7.42717504 4.99731866 -6.3010465 2.62649532 1.95961766 -1.625624652

5 -23.11348501 -5.30321318 -16.2664747 -0.63309346 9.20471867 -3.030887859

6 -6.72047468 -1.02720870 -7.1964647 0.31305547 3.79640492 -1.317280912

7 -3.84695096 -2.14233582 14.4352832 -3.29941875 -7.07853637 1.058475584

8 3.41835391 3.84526835 3.9981480 0.15438064 -3.29535975 -1.418397631

9 -3.76628930 -1.00662862 12.6735898 -2.84775910 -6.54829274 0.280601391

10 1.69713118 2.75088789 9.4842403 -1.16396119 -5.89995272 -0.902241903

11 -2.17155452 -1.49750617 9.8429831 -2.19660259 -4.78843339 0.855596523

12 1.09046323 2.34852142 7.9590057 -1.01462281 -4.98397380 -0.864234692

13 -0.52010274 1.16014349 6.7183091 -1.13174598 -3.99367075 -0.561842111

14 -0.33984571 -0.61639320 8.3597938 -1.60097251 -4.21717323 0.680259005

15 3.30580157 0.20955687 1.1379610 0.24043360 -0.49243200 0.727925611

16 0.32638005 -3.59644338 -1.0598864 -0.17438814 1.86183273 2.309372804

55 56 57 58 59 60

1 2.3076508 -5.24235309 -7.82201221 -5.48857951 -5.92867727 -3.8369272

2 4.1131851 -10.60555704 -5.79532682 -6.70547349 -7.36375824 -2.2886665

3 -2.2338901 1.20796915 0.03438003 -0.93235131 -0.12252922 -0.6160416

4 2.2829695 -1.32860055 -2.26198022 -0.18464569 -1.00310246 -0.6247013

5 20.5674349 -7.57466581 3.51987561 12.82882368 5.02045560 7.9136114

6 7.6672414 -2.63979952 0.98614932 4.74932813 1.82016587 2.7711753

7 -5.6688965 -2.70802457 -1.65266276 -7.10243521 -4.47349124 -2.6794526

8 -0.2775092 -5.16705531 -4.15353561 -5.71030688 -5.08141702 -2.5412160

9 -3.6592563 -4.43236015 -2.48752879 -7.08182531 -5.11808394 -2.5924896

10 -2.5334879 -5.89519410 -4.54254077 -8.17123940 -6.53087144 -3.4481099

11 -4.2845862 -1.35481751 -0.97777329 -4.77013700 -2.85193610 -1.8560741

12 -1.8016450 -5.33550640 -3.93612991 -6.92294959 -5.64282603 -2.8761876

13 -1.2504407 -4.25363886 -2.78586348 -5.20254844 -4.25934292 -2.0397256

14 -4.1521717 -1.01830252 -1.17182715 -4.53198272 -2.70237864 -1.9177712

15 -3.0338199 2.41559605 0.33750782 -0.60276076 0.42001346 -0.6546186

16 -3.8832184 7.50735875 4.17755914 4.01548648 4.88593729 1.3734720

61

1 0.812556234

2 0.278256012

3 0.741244334

4 -0.398882181

5 -7.741920403

6 -2.851056318

7 2.665420660

8 1.216480842

9 2.181458984

10 2.204018717

11 1.910657062

12 1.779501152

13 1.292254389

14 1.860829579

15 0.892369949

16 0.162420760

> #Calculamos el error residual

**> Error\_res <- Xs[, 1:3] - dtorg\_lossinf[,1:3]**

**> Error\_res <- as.matrix(Error\_res)**

**> Error\_res**

rent inc\_sal inc\_ret

1 -19.30583235 -1.79524107 -2.100988298

2 -1.87601524 -10.67939461 1.534284813

3 -0.95447078 1.33018772 -0.891455846

4 -7.51795735 -0.21261498 -0.557978043

5 15.39227952 -8.14101076 7.154365202

6 3.46165403 -3.38616663 2.350323620

7 4.22844578 -3.73690149 -1.011251458

8 -3.59258614 -3.82444908 -0.498112794

9 2.86842310 -5.92456075 0.107123901

10 -2.53293291 -6.31607132 -0.482540537

11 2.54339739 -2.01892348 -0.407344934

12 -2.19034928 -5.92897145 0.089058416

13 0.15508187 -3.91462069 -0.106740582

14 0.94415667 -1.73504415 -0.724814618

15 -2.03673186 2.36722579 -0.915965254

16 2.14875451 7.41807361 -1.227369482

17 -3.57356002 6.50555896 -0.797036545

18 1.91134303 -7.80238910 0.398159693

19 -7.29209266 -8.86964588 0.384743564

20 -3.29381545 -0.54994568 0.387915594

21 7.49119349 4.33687961 -0.989061772

22 14.69894413 -3.89383347 0.720634089

23 3.88800957 2.40733223 -0.571025341

24 4.76736773 8.89659157 -1.366524414

25 6.98248260 9.71555295 -0.911404397

26 1.81445669 -6.14605067 -0.450814276

27 1.51237459 0.16173197 -0.005027534

28 2.67746596 8.38607391 0.281350370

29 -11.33940817 -5.86363150 -0.739556069

30 -5.40315133 -2.66003690 -1.467567063

31 -5.01746971 -1.76363537 -0.710660601

32 7.39940633 3.39242675 -1.076936531

33 9.51434212 3.41817541 -0.328276128

34 2.15005508 -5.45513879 1.503982116

35 -3.93553580 -1.77081544 0.510493805

36 -3.72049853 3.98101163 -0.349954877

37 1.99565971 -0.17673212 1.693762133

38 -0.02402785 1.84892322 0.834380574

39 -1.01664935 2.17153191 0.914662951

40 -2.62039720 0.24897930 1.169285651

41 2.99779809 -1.42677597 1.856637532

42 -1.02091436 -0.05288844 1.498960157

43 -4.68117363 -3.41654990 0.640586482

44 -2.05519342 -1.18568238 0.428640174

45 -0.88553920 4.37252836 -0.230099902

46 -3.57222426 4.87459219 0.149681055

47 3.14559676 5.78140443 -0.911377762

48 0.72320173 0.55078830 -0.197240900

49 -20.40494089 1.88694585 -2.942567573

50 -13.80205852 -4.27391949 -0.765681123

51 10.74204969 -3.11891805 0.062654913

52 -6.51464488 -0.11624633 -0.511963681

53 -1.89065338 3.05391331 0.170506930

54 3.97564707 2.71170865 -0.276685029

55 -3.46042066 -5.53032066 1.681792176

56 5.93684142 12.14480816 -1.746971578

57 7.58432150 5.86733626 -0.021189127

58 4.88912427 6.40956348 0.133621589

59 5.26427384 6.87543218 -0.284063215

60 3.43404218 1.99857885 0.403699045

61 -1.70694528 -1.42672993 -0.485059259

> #Calculamos la desviación típica del error residual.

**> sd(Error\_res)**

[1] 4.801111

**> round(sd(Error\_res), digits = 2)**

[1] 4.8