Reto 4

Paula Corbatón Álvarez

2023-01-05

Pregunta 0. Los resultados del informe corresponden al **SEGUNDO INTENTO**.

```
#cargamos los datos y las librerías que vamos a necesitar para desarrollar los ejercicios
knitr::opts_chunk$set(echo = TRUE)
library(readr)
library(fields)
library(dplyr)
library(MASS)
library(stats)
library(base)
library(graphics)
library(magrittr)
library(corrplot)
gdp <- read_csv("gdp.csv")
```

Pregunta 1:

Creamos la matriz I con los indicadores

```
data_indicators <- indicators[2:9]
I <- data.matrix(data_indicators)
class(I)
```

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

Creamos el vector pib con la columna gdpp del dataframe gdp

```
pib <- gdp$gdpp
class(pib)
```

```
## [1] "numeric"
```

Para saber en cuántos países tiene capacidad de acción la ONG utilizamos el siguiente código que nos indica la longitud del vector pib

```
length(pib)
```

```
## [1] 167
```

Como podemos observar, la ONG tiene capacidad de acción en **167 países**.

Pregunta 2:

Para obtener los 5 primeros paises con menor PIB por cápita podemos ordenar la base de datos gdp según el valor de gdpp y seleccionar las primeras 5 filas. Lo haremos de la siguiente forma:

```
sorted_gdp <- gdp %>%
  arrange(gdpp) %>%
  head(5)
sorted_gdp
```

```
## # A tibble: 5 × 2
##   region            gdpp
##   <chr>            <dbl>
## 1 Burundi            231
## 2 Liberia            327
## 3 Congo, Dem. Rep.   334
## 4 Niger              348
## 5 Sierra Leone       399
```

Como podemos observar, los **países ordenados de menor a mayor** son:

1. Burundi
2. Liberia
3. Congo, Dem. Rep.
4. Niger
5. Sierra Leone

Pregunta 3:

Normalizamos los datos de la matriz I

```
Is <- scale(I)
```

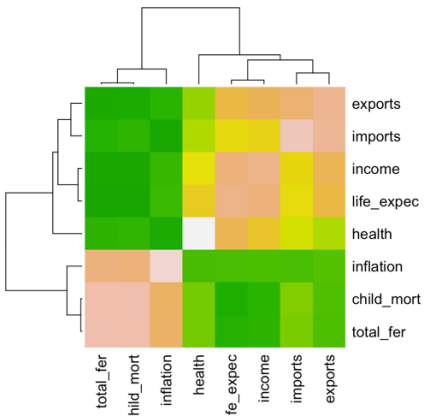
Calculamos la matriz de covarianza

```
CIs <- cov(Is)
CIs
```

```
##             child_mort    exports      health     imports     income  inflation   life_expec  total_fer
## child_mort   1.0000000 -0.3180932 -0.20040206 -0.12721092 -0.5243150  0.2882762 -0.88667610  0.8484781
## exports     -0.3180932  1.0000000 -0.11440840  0.73738083  0.5167836 -0.1072944  0.31631260 -0.3200106
## health      -0.2004021 -0.1144084  1.00000000  0.09571668  0.1295786 -0.2553758  0.21069212 -0.1966740
## imports     -0.1272109  0.7373808  0.09571668  1.00000000  0.1224062 -0.2469943  0.05439053 -0.1590484
## income      -0.5243150  0.5167836  0.12957861  0.12240625  1.0000000 -0.1477560  0.61196247 -0.5018401
## inflation    0.2882762 -0.1072944 -0.25537579 -0.24699428 -0.1477560  1.0000000 -0.23970496  0.3169211
## life_expec -0.8866761  0.3163126  0.21069212  0.05439053  0.6119625 -0.2397050  1.00000000 -0.7608747
## total_fer   0.8484781 -0.3200106 -0.19667399 -0.15904843 -0.5018401  0.3169211 -0.76087469  1.0000000
```

Dibujamos la matriz de covarianza

```
heatmap(cor(CIs))
```



Como podemos observar, los **indicadores que están más relacionados** en valor absoluto (aquellos cuyo valor se acerca más a 1) son: **life_expec y child_mort** con una covarianza de -0.8866761. Por otra parte, los **indicadores que están menos relacionados** (aquellos cuyo valor se acerca más a 0) son: **life_expec e imports** con una covarizanza de 0.05439053

Pregunta 4:

Primero, llevamos a cabo el análisis de componentes principales y lo asignamos a una variable llamada “pca_results”:

```
pca_results <- prcomp(I, scale = TRUE) #scale = TRUE estandariza los datos antes de llevar a cabo el PCA
summary(pca_results)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8
## Standard deviation     1.8907 1.2426 1.0786 0.85953 0.74980 0.47274 0.32945 0.29162
## Proportion of Variance 0.4468 0.1930 0.1454 0.09235 0.07028 0.02794 0.01357 0.01063
## Cumulative Proportion  0.4468 0.6398 0.7852 0.87759 0.94787 0.97580 0.98937 1.00000
```

Como podemos ver en la fila “Proportion of Variance” las componentes que explican por sí mismas al menos un 10% de la varianza total son:

- PC1
- PC2
- PC3

Por el enunciado entiendo que la proyeccion de los datos se tiene que hacer con los componentes que explican por si mismos el 10% de la varianza total. Por to tanto, extraigo los componentes principales (PC1, PC2 y PC3) de la variable pca_results:

```
principal_components <- pca_results$rotation[,0:3]
```

Proyectamos los datos en los componentes principales:

```
projected_data <- Is %*% principal_components # con el operador '%*%' estamos llevando a cabo la multiplicación d
e matrices (Is * principal_components)

# Definimos la proyección de los datos sobre cada una de las componentes
index1 <- projected_data[, 1]
index2 <- projected_data[, 2]
index3 <- projected_data[, 3]
head(projected_data, n=5)
```

```
##              PC1         PC2         PC3
## [1,] -2.89717025  0.15783911 -0.90634204
## [2,]  0.72190125 -0.65421607 -0.11221647
## [3,] -0.09651877 -0.47641205  1.35647114
## [4,] -2.99479252  1.78253063  1.30721288
## [5,]  1.17690487  0.08360425 -0.06929212
```

4.a) Calcular los valores y vectores propios de la matriz de covarianza.

```
# Calculamos los valores propios y los vectores propios de la matriz de covarianza "CIs" que creamos en el ejerci
cio 3
eigen_results <- eigen(CIs)

# Extraemos los valores y los vectores propios y los asignamos a las variables "eigenvalues" y "eigenvectors" res
pectivamente
eigenvalues <- eigen_results$values
eigenvectors <- eigen_results$vectors
eigenvalues
```

```
## [1] 3.57462318 1.54394608 1.16337336 0.73879126 0.56220111 0.22348725 0.10853621
## [8] 0.08504155
```

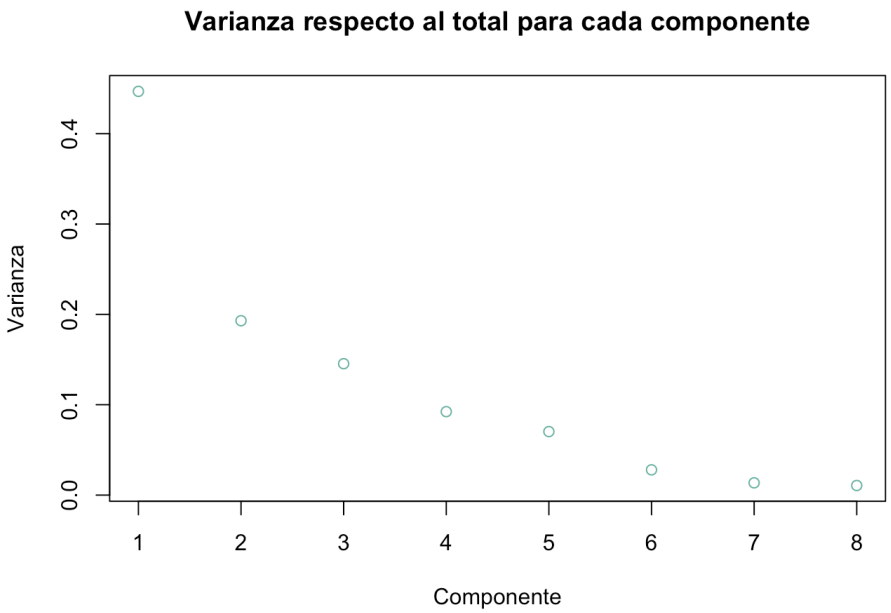
```
eigenvectors
```

```
##            [,1]         [,2]        [,3]        [,4]        [,5]        [,6]        [,7]         [,8]
## [1,]  0.4728799 -0.214124060  0.09998804 0.115186553 -0.29716975  0.20332080  0.13513304  0.747903607
## [2,] -0.3083961 -0.608374217 -0.14603735 0.101508207 -0.05751086 -0.05344747  0.69641863 -0.109447681
## [3,] -0.1445682  0.241608166  0.64740271 0.680155939  0.05895877  0.01392064  0.18267289 -0.044089348
## [4,] -0.1946400 -0.661131277  0.28525732 0.056360711  0.31536808 -0.03654309 -0.56924463  0.125062238
## [5,] -0.3867871 -0.031206520 -0.24777586 0.315028674 -0.72825615  0.17896303 -0.35135794 -0.054302761
## [6,]  0.2204750 -0.005770746 -0.61577680 0.621291977  0.41786462  0.06357684 -0.08615010  0.009899599
## [7,] -0.4641913  0.237343411 -0.15808191 0.003856991  0.09136627 -0.60043527  0.02034424  0.577845597
## [8,]  0.4569516 -0.176701966  0.05108475 0.159304266 -0.30353554 -0.74678145 -0.08968381 -0.272258407
```

4.b) Dibujar la gráfica de la varianza respeto al total (eje de ordenadas) para cada componentes (eje de abscisas)

```
# Extraemos la varianza explicada de cada componente
explained_variance <- summary(pca_results)$importance[2,]

# Dibujamos la gráfica
plot(explained_variance, xlab = "Componente", ylab = "Varianza", main = "Varianza respecto al total para cada com
ponente", col="#69b3a2")
```



4.c) Calcular Nc, el número de componentes que superan el 10% de la varianza.

```
# Contamos el número de componentes cuya varianza explicada es igual o mayor al 10%
Nc <- sum(explained_variance >= 0.1)

# Mostramos el resultado en pantalla
print(paste("Hay", Nc, "componentes cuya varianza explicada es igual o mayor al 10%"))
```

```
## [1] "Hay 3 componentes cuya varianza explicada es igual o mayor al 10%"
```

Como nos muestra el código hay **3 componentes** cuya varianza explicada es igual o mayor al 10%. Estos componentes son los detallados al principio del ejercicio:

- PC1
- PC2
- PC3

4.d) Calcular la varianza acumulada, respeto del total, con las Nc componentes

```
summary(pca_results)
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6     PC7     PC8
## Standard deviation     1.8907 1.2426 1.0786 0.85953 0.74980 0.47274 0.32945 0.29162
## Proportion of Variance 0.4468 0.1930 0.1454 0.09235 0.07028 0.02794 0.01357 0.01063
## Cumulative Proportion  0.4468 0.6398 0.7852 0.87759 0.94787 0.97580 0.98937 1.00000
```

La **varianza acumulada**, en porcentaje redondeado a 2 decimales es: **78.52%**

Pregunta 5:

Para calcular la proyección de los datos sobre cada una de las componentes principales podemos utilizar el código del ejercicio 4:

```
principal_components <- pca_results$rotation #extraigo los componentes principales

#Proyectamos los datos en los componentes principales:
projected_data <- Is %*% principal_components # con el operador '%*%' estamos llevando a cabo la multiplicación d
e matrices (Is * principal_components)

row.names(projected_data) <- indicators$region #establecemos como nombre de la columna, la columna region (de la
base de datos indicators)

projected_data["Brazil", ] #elegimos la fila con los datos correspondientes a Brazil como nos pide el enunciado
```

```
##         PC1         PC2         PC3         PC4         PC5         PC6
##  0.17839578 -2.15942216 -0.11380722  0.17077312  0.18224482  0.32972716
##         PC7         PC8
##  0.25599066 -0.01368224
```

Como podemos observar la proyección sobre la primera componente principal de **Brazil** (redondeando a 2 decimales) es: **-0.18** (lo pongo en negativo como se ha indicado en el foro)

Pregunta 6:

Para calcular la desviación estándar del error residual que obtenemos usando las Nc componentes haremos lo siguiente:

```
residual_error <- pca_results$sdev[4:8] #extraemos los residuales del análisis de componentes principales
sd(residual_error) # calculamos la desviación estándar
```

```
## [1] 0.2533091
```

La desviación estándar del error residual (redondeando a dos decimales) es: **0.25**

Pregunta 7:

Para estudiar qué variable ("indicador") influye más en la componente ("indice") PC3 podemos utilizar la siguiente tabla:

```
pca_results$rotation[,3]
```

```
##  child_mort     exports      health     imports      income   inflation
## -0.09998804  0.14603735 -0.64740271 -0.28525732  0.24777586  0.61577680
##  life_expec   total_fer
##  0.15808191 -0.05108475
```

Así podemos ver que para la PC3 la variable ("indicador") que más influye es **health** (-0.64740271) seguida de **inflation** (0.61577680)

Pregunta 8:

Para calcular la covarianza del índice 2 con el PIB por cápita haremos lo siguiente:

```
components <- pca_results$x[, 0:Nc] #extraemos los componentes principales
# el valor del pib por cápita ya lo tenemos guardado en la variable "pib"

#estandarizamos los valores
components_std <- scale(components)
pib_std <- scale(pib)

#Calculamos la matriz de covarianza
cov_matrix <- cov(components_std, pib_std)
cov_matrix
```

```
##            [,1]
## PC1  0.69379370
## PC2 -0.04189602
## PC3  0.05196479
```

```
indices_counties <- data.frame(projected_data)
row.names(indices_counties) <- indicators$region

sorted_indicators_by_PC <- indices_counties %>%
  arrange(PC3) %>%
  head(5)
sorted_indicators_by_PC
```

```
##                             PC1         PC2       PC3       PC4         PC5
## Micronesia, Fed. Sts. -0.173102  0.07319759 -2.758727 1.4396309  0.8354823
## Lesotho               -1.661374  2.04626577 -2.649562 0.9278294  0.4442159
## Liberia               -1.632512  1.06667514 -2.546481 1.2140570  0.3867302
## Kiribati              -1.060273  0.35832568 -2.422764 0.6236862  0.4592499
## US                     1.889963 -2.91313246 -2.110853 2.5153200 -1.0687447
##                             PC6        PC7        PC8
## Micronesia, Fed. Sts. -0.02760339 -0.5105967  0.2318600
## Lesotho                1.54465877 -0.6021427  0.2313731
## Liberia               -0.27358436 -0.9579838 -0.2279770
## Kiribati               0.19490558 -0.8460042  0.1006258
## US                     0.21519711  0.1795649  0.1829755
```