GP. Assignment –The Covariance Matrix / Multidimensional Scaling PSYP14 (HT2020)

Karima Kandi

2020-12-09

#loading necessary packages  
suppressMessages(suppressWarnings(  
 library(psych, quietly = TRUE, warn.conflicts = FALSE)))  
suppressMessages(suppressWarnings(  
 library(lsr, quietly = TRUE, warn.conflicts = FALSE)))  
suppressMessages(suppressWarnings(  
 library(tidyverse, quietly = TRUE, warn.conflicts = FALSE)))  
suppressMessages(suppressWarnings(  
 library(readr, quietly = TRUE, warn.conflicts = FALSE)))  
suppressMessages(suppressWarnings(  
 library(sciplot, quietly = TRUE, warn.conflicts = FALSE)))  
suppressMessages(suppressWarnings(  
 library(reshape2, quietly = TRUE, warn.conflicts = FALSE)))  
suppressMessages(suppressWarnings(  
 library(MASS, quietly = TRUE, warn.conflicts = FALSE)))  
suppressMessages(suppressWarnings(  
 library(smacof, quietly = TRUE, warn.conflicts = FALSE)))  
suppressMessages(suppressWarnings(  
 library(lm.beta, quietly = TRUE, warn.conflicts = FALSE)))  
suppressMessages(suppressWarnings(  
 library(gridExtra, quietly = TRUE, warn.conflicts = FALSE)))  
suppressMessages(suppressWarnings(  
 library(magrittr, quietly = TRUE, warn.conflicts = FALSE)))  
suppressMessages(suppressWarnings(  
 library(dplyr, quietly = TRUE, warn.conflicts = FALSE)))

### Assignment 1, part 1

“Try writing a short R function to calculate a (Pearson) covariance matrix for the womenshealth.txt data set. Ideally, your function will accept any number of variables (> 2, no need to test for missing values) and return a covariance matrix for the variables.”

Importing the data file and extracting relevant variables

womenshealth <- read.delim("C:/Users/am3451kl/Dropbox/PhD/Undervisning/Pedagogiskt stöd/womenshealth.txt", header = TRUE)  
sum(is.na(womenshealth))

## [1] 26

Note, the dataset womenshealth includes 26 missing values.  
To make calculations easier henceforth, I will only variables in columns 1-4.

womenshealth.short = womenshealth[1:4]  
  
view(womenshealth.short)

### Writing function for calculation of covariance matrix

Extracting variables into numeric vectors

vec1 <- as.numeric(womenshealth.short$subno)  
vec2 <- as.numeric(womenshealth.short$timedrs)  
vec3 <- as.numeric(womenshealth.short$attdrug)  
vec4 <- as.numeric(womenshealth.short$atthouse)

Equation for calculating the covariance between x and y

Covariance(x,y) = (1/N-1) × (Σ(x - ) × (y - ))

#### Covariance matrix function for two numeric vectors

my.cov1 <- function(vec1, vec2) {  
 covariance <- sum((vec1-mean(vec1))\*(vec2-mean(vec2)))/(NROW(vec1)-1)   
 return(covariance)  
}  
print(my.cov1(vec1,vec2))

## [1] 40.46809

Double checking that my function provides the same answer as the function in base R

cov(vec1,vec2)

## [1] 40.46809

Conclusion: my covariance function returns the same answer as base R for the variables subno and timedrs.

### Write up in the style of a ? vignette

Description

my.cov(x,y) calculates the covariance matrix of two numeric vectors.

Arguments x a numeric vector y a numeric vector

Values Returns the covariance matrix of x and y

### Scaling covariance matrix into correlation

“If you have computed the covariance matrix successfully, try taking a step further by writing another short function to scale the covariance matrix into correlation.”

Equation for calculating the correlation between x and y from the covariance of x and y

Correlation(x,y) = covariancexy / (SDx × SDy)

my.corr1 <- function(vec1,vec2) {  
 corr.calc <- my.cov1(vec1,vec2)/(sd(vec1)\*sd(vec2))  
 return(corr.calc)}  
  
  
print(my.corr1(vec1, vec2))

## [1] 0.01897131

Double checking that my function provides the same answer as the function in base R

print(cor(vec1, vec2))

## [1] 0.01897131

Conclusion: my correlation function returns the same answer as base R for the variables subno and timedrs.

### Expansion of the function for covariance matrix to accept all variables in womenshealth.short

my.cov2 <- function(vec1, vec2, vec3, vec4) {  
 covariance12 <- sum((vec1-mean(vec1))\*(vec2-mean(vec2)))/(NROW(vec1)-1)  
 covariance13 <- sum((vec1-mean(vec1))\*(vec3-mean(vec3)))/(NROW(vec1)-1)  
 covariance14 <- sum((vec1-mean(vec1))\*(vec4-mean(vec4)))/(NROW(vec1)-1)  
 covariance23 <- sum((vec2-mean(vec2))\*(vec3-mean(vec3)))/(NROW(vec2)-1)  
 covariance24 <- sum((vec2-mean(vec2))\*(vec4-mean(vec4)))/(NROW(vec2)-1)  
 covariance34 <- sum((vec3-mean(vec3))\*(vec4-mean(vec4)))/(NROW(vec3)-1)  
 return(matrix(c(covariance12, covariance13, covariance14, covariance23, covariance24, covariance34), nrow = 6, ncol = 1))  
}  
  
(my.cov2(vec1,vec2,vec3, vec4))

## [,1]  
## [1,] 40.4680935  
## [2,] 2.8967303  
## [3,] -74.2953796  
## [4,] 1.3339364  
## [5,] 5.7722025  
## [6,] 0.1457244

This is the covariance matrix for variables subno, timedrs, atthouse, attdrug. It is written in the order cov timedrs-subno, attdrug-subno, atthouse-subno, timedrs-attdrug, timedrs-atthouse.

Double checking that my function provides the same answer as the function in base R

cov(womenshealth.short, use = "pairwise.complete.obs")

## subno timedrs attdrug atthouse  
## subno 37843.61631 40.468093 2.8967303 -74.2953796  
## timedrs 40.46809 120.237020 1.3339364 5.7722025  
## attdrug 2.89673 1.333936 1.3419072 0.1457244  
## atthouse -74.29538 5.772203 0.1457244 18.1280945

Conclusion: my covariance function returns the same answer as base R for the variables subno,timedrs, attdrug, and atthouse.

### Scaling the expanded covariance matrix into a correlation matrix

#Fix this before next time (not working right now)  
my.corr2 <- function(vec1,vec2,vec3, vec4) {  
 corr.calc12 <- my.cov1(vec1, vec2)/(sd(vec1)\*sd(vec2))  
 corr.calc13 <- my.cov1(vec1, vec3)/(sd(vec1)\*sd(vec3))  
 corr.calc14 <- my.cov1(vec1, vec4)/(sd(vec1)\*sd(vec4))  
 corr.calc23 <- my.cov1(vec2, vec3)/(sd(vec2)\*sd(vec3))  
 corr.calc24 <- my.cov1(vec2, vec4)/(sd(vec2)\*sd(vec4))  
 corr.calc34 <- my.cov1(vec3, vec4)/(sd(vec3)\*sd(vec4))  
 return(matrix(c(corr.calc12, corr.calc13, corr.calc14, corr.calc23, corr.calc24, corr.calc34), nrow = 6, ncol = 1))}  
  
  
print(my.corr2(vec1, vec2, vec3, vec4))

## [,1]  
## [1,] 0.01897131  
## [2,] 0.01285436  
## [3,] -0.08969937  
## [4,] 0.10501583  
## [5,] 0.12363641  
## [6,] 0.02954575

Double checking that my function provides the same answer as the function in base R

print(cor(womenshealth.short, use = "pairwise.complete.obs"))

## subno timedrs attdrug atthouse  
## subno 1.00000000 0.01897131 0.01285436 -0.08969937  
## timedrs 0.01897131 1.00000000 0.10501583 0.12363641  
## attdrug 0.01285436 0.10501583 1.00000000 0.02954575  
## atthouse -0.08969937 0.12363641 0.02954575 1.00000000

Conclusion: my correlation function returns the same answer as base R for the variables subno,timedrs, attdrug, and atthouse.

### Assignment 1, part 2: Non-metric multidimensional scaling

“Following the procedures discussed by Everitt and Hothorn (2011), use multidimensional scaling to examine the students’ perceived dissimilarities between the nations.”

The dataset Nations represent ratings of global similarity of different pairs of nations, conducted by 18 students before 1970 on a scale from 1=very different' to9=very similar’.

Nations <- read.delim("C:/Users/am3451kl/Dropbox/PhD/Undervisning/Pedagogiskt stöd/Nations.txt")  
View(Nations)

Dissimilarity matrix

#calculating dissimilarity ratings from the similarity ratings by inverting the scale  
Nat.D = sim2diss(Nations, method = 10)   
  
print(Nat.D)

## Brazil Congo Cuba Egypt France India Israel Japan China UdSSR USA  
## [1,] 10.00 5.17 4.72 6.56 5.28 5.50 6.17 6.50 7.61 6.94 4.61  
## [2,] 5.17 10.00 5.44 5.00 6.00 5.17 6.67 6.61 6.00 6.61 7.61  
## [3,] 4.72 5.44 10.00 4.83 5.89 6.00 6.39 7.06 4.50 4.56 6.83  
## [4,] 6.56 5.00 4.83 10.00 5.22 4.17 5.33 6.17 5.61 5.61 6.67  
## [5,] 5.28 6.00 5.89 5.22 10.00 6.56 6.00 5.78 6.33 4.94 4.06  
## [6,] 5.50 5.17 6.00 4.17 6.56 10.00 5.89 5.50 5.89 5.50 5.72  
## [7,] 6.17 6.67 6.39 5.33 6.00 5.89 10.00 5.17 7.00 5.83 4.06  
## [8,] 6.50 6.61 7.06 6.17 5.78 5.50 5.17 10.00 5.83 5.39 3.94  
## [9,] 7.61 6.00 4.50 5.61 6.33 5.89 7.00 5.83 10.00 4.28 7.44  
## [10,] 6.94 6.61 4.56 5.61 4.94 5.50 5.83 5.39 4.28 10.00 5.00  
## [11,] 4.61 7.61 6.83 6.67 4.06 5.72 4.06 3.94 7.44 5.00 10.00  
## [12,] 6.83 6.50 4.89 5.72 5.28 6.00 5.56 5.72 4.94 3.33 6.44  
## Yugoslavia  
## [1,] 6.83  
## [2,] 6.50  
## [3,] 4.89  
## [4,] 5.72  
## [5,] 5.28  
## [6,] 6.00  
## [7,] 5.56  
## [8,] 5.72  
## [9,] 4.94  
## [10,] 3.33  
## [11,] 6.44  
## [12,] 10.00

### Running non-metric multidimensional scaling

#transforming dissimilarity ratings into a dissimilarity distance matrix  
Nat.Ddist <- as.dist(Nat.D)  
#running the actual multidimensional scaling  
(N.mds = isoMDS(Nat.Ddist))

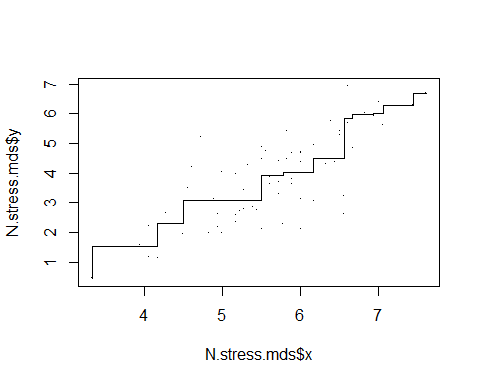
## initial value 22.139081   
## iter 5 value 19.057768  
## final value 18.855420   
## converged

## $points  
## [,1] [,2]  
## Brazil 1.7601751 -2.9725809  
## Congo -2.2095850 -3.0421037  
## Cuba -2.7535795 -0.3547153  
## Egypt -1.0716881 -1.4225984  
## France 0.9691242 0.3820645  
## India -0.1108395 -2.0073935  
## Israel 3.0024371 -0.1595721  
## Japan 2.2058681 2.2918957  
## China -3.1587746 1.5626636  
## UdSSR -0.6589934 2.4348061  
## USA 3.1061716 1.0144607  
## Yugoslavia -1.0803160 2.2730733  
##   
## $stress  
## [1] 18.85542

According to the rule of thumb given in Everitt & Hothorn (2011, p.123) stress values ≥ 20%, poor, Stress = 10%, fair, Stress ≤ 5%, good, Stress = 0, perfect.The stress for this MDS solution is 18.86%, thus indicating fair to poor fit in 2 dimensions.

Running Shephard’s plot to make sense of the stress value.

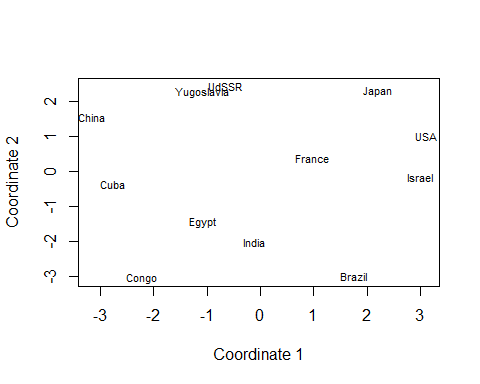
N.stress.mds <- Shepard(Nat.Ddist, N.mds$points)  
{plot(N.stress.mds, pch = ".")  
lines(N.stress.mds$x, N.stress.mds$yf, type = "S")  
}



Conclusion: Fair to poor, closer to poor fit, also indicated by Shephard’s plot given the non-linear shape of the line and the distance of the data points from the line.

Plotting the two-dimensional MDS solution

x2 <- N.mds$points[,1]  
y2 <- N.mds$points[,2]  
  
### Making a simple plot  
{plot(x2, y2, xlab = "Coordinate 1",   
 ylab = "Coordinate 2",   
 type = "n")  
text(x2, y2,   
 labels = colnames(Nations),   
 cex = 0.7)  
}



### Analysis of coordinates in two-dimensions

The closest two nations in this plot are Yugoslavia and UdSSR, followed by the USA and Israel. Egypt and India also place rather close to each other. One interpretation could be that that the dimension on the x-axis represent similarity in terms of economic system with placements further to the left indicating a more communist economy, and placements further to the right indicate a more capitalist economy. The dimension on the y-axis on the other hand seem to represent similarity in terms of relationship to colonisation. Placements higher up on the y-axis include nations that have had colonies while placements lower down on the y-axis include nations that have been colonies. Note though that the fit of the model in two dimensions was only fair to poor.