MDS Assessment

Roger Bukuru

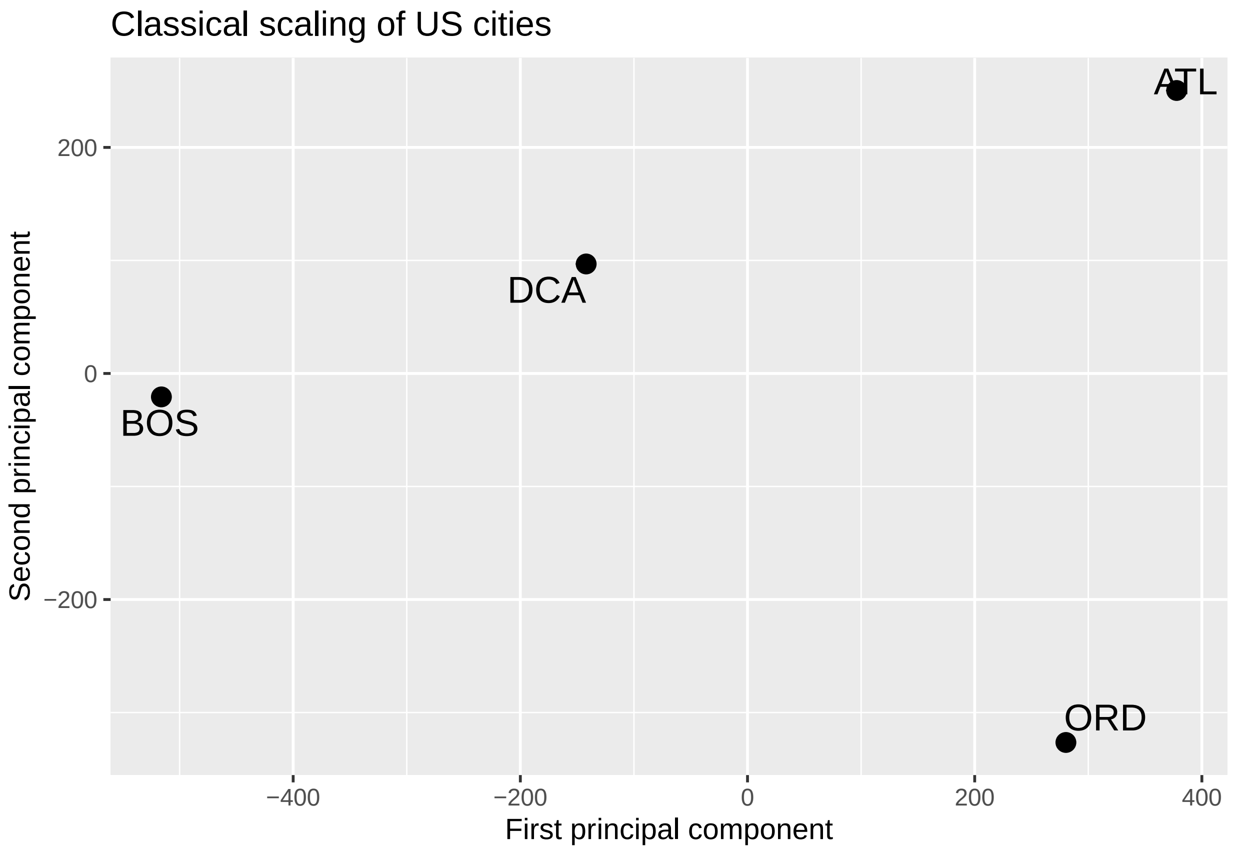
2024-02-17

Question 1

For this exercise we will be using the cities data that is found in the psych package in R. Run the following lines of code to obtain the dissimilarity matrix that we will work with.

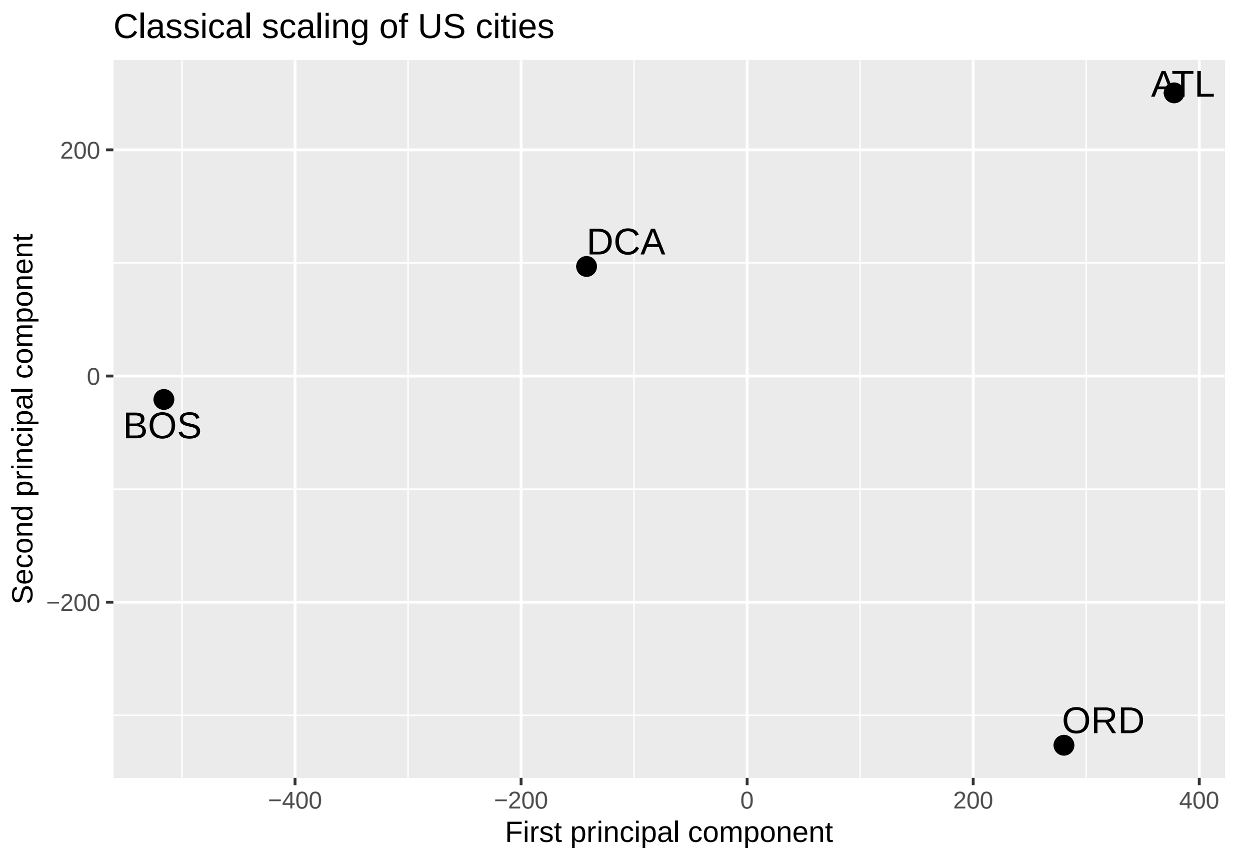
a) Using the first 4 (four) cities, ATL, BOS, ORD and DCA. Perform the classical scaling algorithm by first principals to obtain the principals coordinates. Plot the principal coordinates in 2 dimensions. You can use R to create matrices and perform the Eigen value decomposition.

```
rm(list = ls())
library(psych)
library(psychTools)
library(tibble)
library(dplyr)
library(ggplot2)
library(ggrepel)
data(cities)

classical_scaling_first_principals = function(noOfCities = 4) {
    proximity_matrix = as.matrix(cities[1:noOfCities, 1:noOfCities])

    # A = -0.5 * sqrt(proximity_matrix)
    A = (-1/2) * proximity_matrix^2

    # B = HAH H = I-1/n11^1
    n = nrow(A)
    H = diag(1, nrow = n) - 1/n * matrix(1, nrow = n, ncol = ncol(A))
    B = H %*% A %*% H

    # Objective function Y = Eigenvectors %*%
    # diag(eigenvalues)
    evd = eigen(B)
    Y = evd$vectors %*% diag(sqrt(evd$values))  # Principal components

    principal_comp = Y %>%
        as.tibble() %>%
        mutate(city = colnames(proximity_matrix))
    # First two components, represent comps with the most
    # variance in representing the data with comp 1 having
    # the most variance etc.
    two_dim_plot = ggplot(principal_comp, mapping = aes(V1, V2)) +
        geom_point(size = 3) + geom_text_repel(aes(label = city),
        size = 5) + labs(x = "First principal component", y = "Second principal component",
        title = "Classical scaling of US cities")
    coord_equal()

    two_dim_plot

}
classical_scaling_first_principals()
```



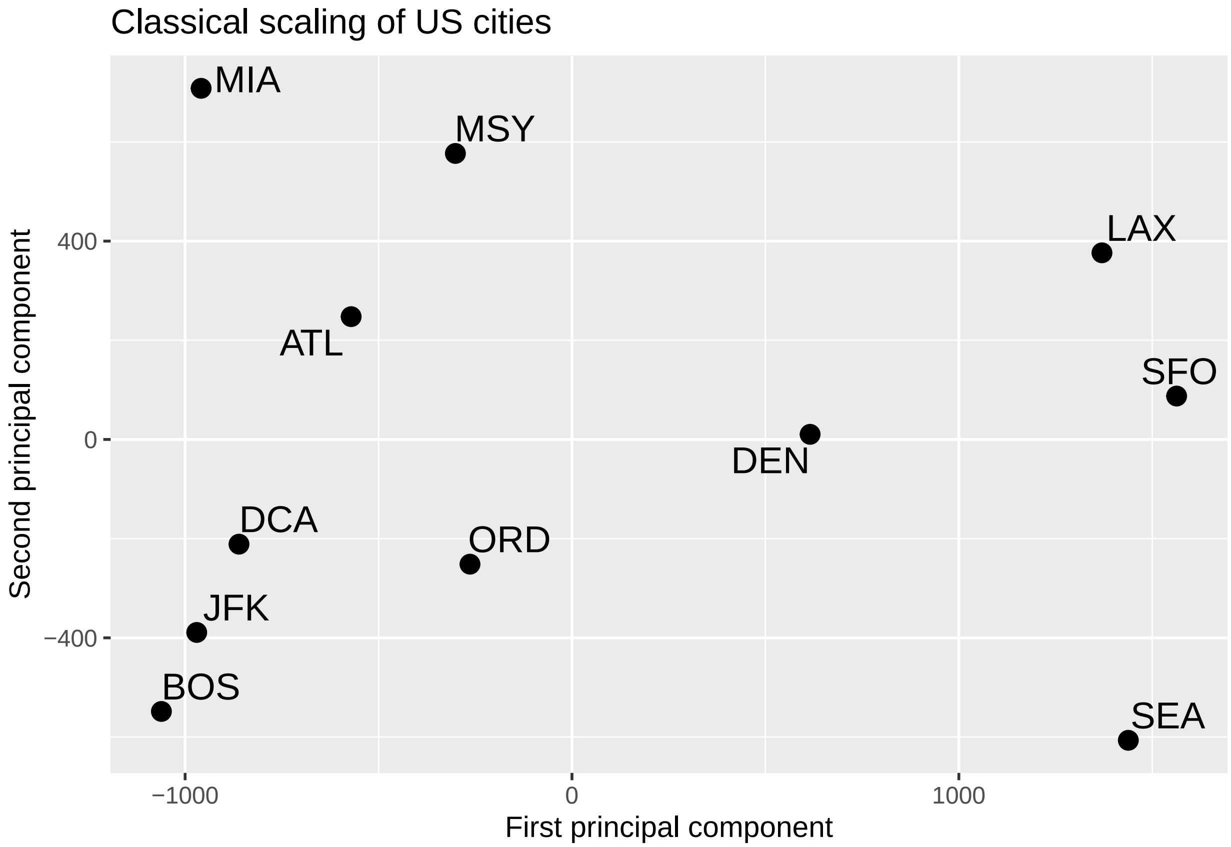
b) Use the function cmdscale in R to check check your answer in part (a).

```
proximity_matrix = as.matrix(cities[1:4, 1:4])
evd = cmdscale(proximity_matrix, eig = TRUE, list. = TRUE)
Y = evd$points
principal_comp = Y %>%
    as.tibble() %>%
    mutate(city = colnames(proximity_matrix))
ggplot(principal_comp, aes(V1, V2)) + geom_point(size = 3) +
    geom_text_repel(aes(label = city), size = 5) + labs(x = "First principal component",
    y = "Second principal component", title = "Classical scaling of US cities") +
    coord_equal()
```



c) Perform the classical scaling algorithm to the entire dissimilarity matrix and obtain the principal coordinates. Plot the resulting coordinates in 2 dimensions.

```
classical_scaling_first_principals(noOfCities = 11)
```



Classical scaling of US cities

Question 2

a) (i) Use the mds() function - in the smacof package - to perform metric multidimensional scaling. Loc

(ii) Which of the 11 cities has the poorest fit? Explain your answer.

(iii) Plot the 2D multidimensional scaling configuration and compare this the locations of the citie

b) Repeat the analysis in part (a) using a random starting solution. How does the different initial configuration change the location of the cities in the 2D configuration?