Multidimensional scaling (MDS)

Humaun Farid Sohag

2025-08-13

Part(a)

Step 1: Create an empty matrix and define its dimensions

```r
# The matrix will hold the distance data between 10 U.S. cities.
Air_line_data_2 <- matrix(ncol = 10, nrow = 10)

# Assign city names to the columns and rows
colnames(Air_line_data_2) <- c("Atlanta", "Chicago", "Denver", "Houston", "Los_Angeles", "Miami", "New_
rownames(Air_line_data_2) <- c("Atlanta", "Chicago", "Denver", "Houston", "Los_Angeles", "Miami", "New_
```

Step 2: Populate the matrix with distance data

```r
# The distances are fed into the lower triangle of the matrix.
Air_line_data_2[lower.tri(Air_line_data_2)] <- c(587, 1212, 701, 1936, 604, 748, 2139, 2182, 543, 920, 

# Set the diagonal to zero (distance from a city to itself is 0)
diag(Air_line_data_2) <- 0
```

Step 3: Convert the matrix to a distance object

```r
# The cmdscale function works with distance objects.
# We also reflect the lower triangle to the upper triangle to make it a full symmetric matrix.
dist_data <- as.dist(Air_line_data_2)
```

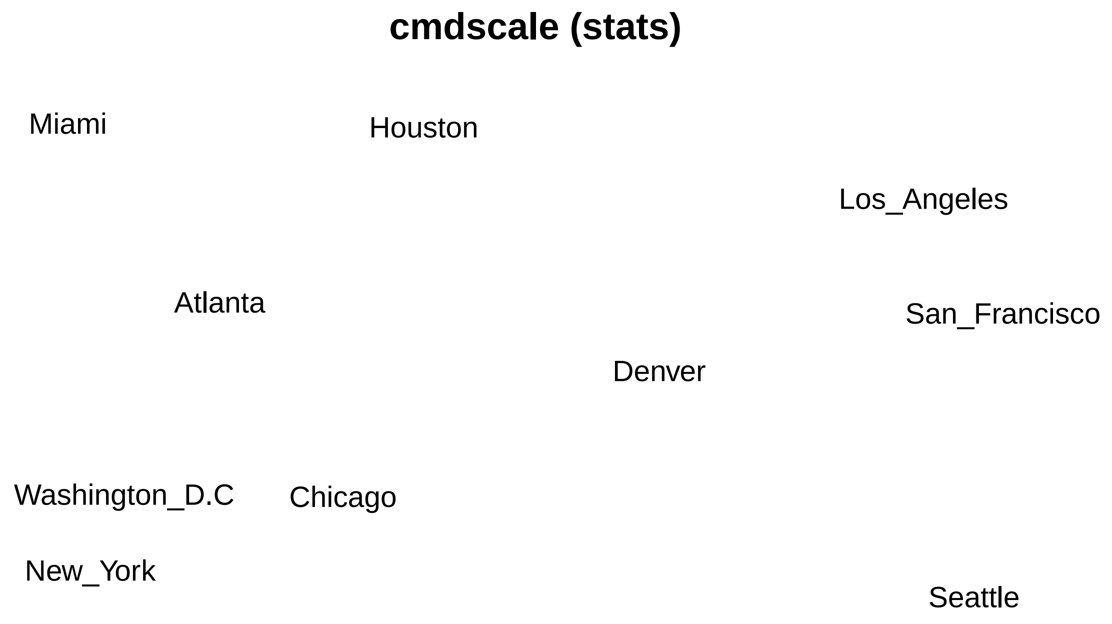
Step 4: Perform Classical (Metric) MDS

```r
# cmdscale() performs the scaling, reducing the data to k=2 dimensions.
MMDS_1 <- cmdscale(dist_data, k = 2)
```

Step 5: Plot the results

```
# 'type="n"' creates an empty plot.
plot(MMDS_1[, 1], MMDS_1[, 2], type = "n", xlab = "", ylab = "", axes = FALSE, main = "cmdscale (stats)

# 'text()' adds the city names to the plot at their new 2D coordinates.
text(MMDS_1[, 1], MMDS_1[, 2], labels = rownames(MMDS_1), cex = 0.9, xpd = TRUE)
```



Part(b)

Step 1: Load the MASS library for the isoMDS function

```
library(MASS)
```

Step 2: Create an empty matrix for the dissimilarity data

```
World_war_Politicians_data_1 <- matrix(ncol = 12, nrow = 12)

# Assign names to the columns and rows
politician_names <- c("Hitler", "Mussolini", "Churchill", "Eisenhower", "Stalin", "Attlee", "Franco", "
colnames(World_war_Politicians_data_1) <- politician_names
rownames(World_war_Politicians_data_1) <- politician_names
```

Step 3: Populate the matrix with dissimilarity data

```
# Fill the lower triangle of the matrix with the provided dissimilarity scores.
World_war_Politicians_data_1[lower.tri(World_war_Politicians_data_1)] <- c(5, 11, 15, 8, 17, 5, 10, 16,

# Set the diagonal to zero
diag(World_war_Politicians_data_1) <- 0
```

Step 4: Convert the matrix to a distance object

```
dist_politicians <- as.dist(World_war_Politicians_data_1)
```

Step 5: Perform Non-Metric MDS

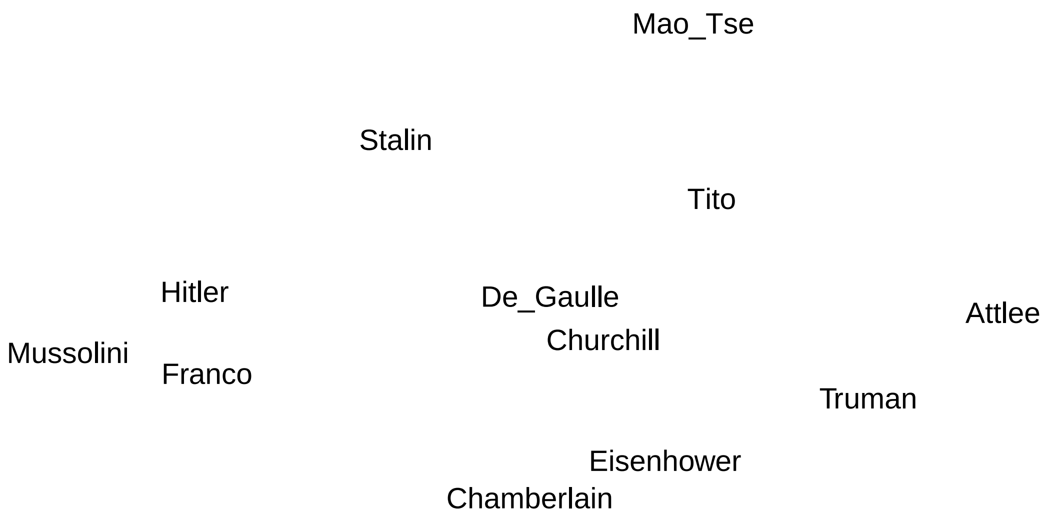
```
# The isoMDS() function is used for non-metric scaling.
NMMDS_1 <- isoMDS(dist_politicians, k = 2)
```

```
## initial  value 18.887607
## iter   5 value 14.915153
## iter  10 value 12.972441
## final  value 12.927660
## converged
```

Step 6: Plot the results

```
# Create an empty plot to place the labels on.
plot(NMMDS_1$points, type = "n", xlab = "", ylab = "", axes = FALSE)

# Add the politician names at their scaled coordinates.
text(NMMDS_1$points, labels = politician_names, cex = 0.9, xpd = TRUE)
```


Mao_Tse
Stalin
Tito
Hitler
De_Gaulle
Attlee
Churchill
Mussolini
Franco
Truman
Eisenhower
Chamberlain