Matrix Example

Arturo Ortiz

2023-10-23

{r setup, include=FALSE} knitr::opts\_chunk$set(echo = TRUE)

## R Markdown

This is an R Markdown document. Markdown is a simple formatting syntax for authoring HTML, PDF, and MS Word documents. For more details on using R Markdown see <http://rmarkdown.rstudio.com>.

When you click the **Knit** button a document will be generated that includes both content as well as the output of any embedded R code chunks within the document. You can embed an R code chunk like this:

summary(cars)

## speed dist   
## Min. : 4.0 Min. : 2.00   
## 1st Qu.:12.0 1st Qu.: 26.00   
## Median :15.0 Median : 36.00   
## Mean :15.4 Mean : 42.98   
## 3rd Qu.:19.0 3rd Qu.: 56.00   
## Max. :25.0 Max. :120.00

## Including Plots

You can also embed plots, for example:



Note that the echo = FALSE parameter was added to the code chunk to prevent printing of the R code that generated the plot.

# Problem 1

# Part A

## There are n rows in an n x 3 matrix. There are Three columns in a n X 3 matrix.

##Below is an example of a 4 x 3 matrix.

n <- 4 A <- matrix(c(4, 6, 2, -1, 0, 3, 7, 1, 8, -5, 6, 10), nrow = n, ncol = 3, byrow = TRUE) print(A)

# Part B

## Below is a 3 x2 matrix.

B <- matrix(c(6, 2, 1, -5, 1, 3), nrow = 3, ncol = 2) print(B)

# Problem 2

# Part A

## The expression is solvable. See below

matrix1 <- matrix(c(2, -1, 1, 2), nrow = 2, byrow = TRUE) matrix2 <- matrix(c(1, 1, 2, 2, 0, 1), nrow = 2, byrow = TRUE) result <- matrix1 %\*% matrix2 print(result)

# Part B

## The expression is solvable. See below

matrixb\_1 <- matrix(c(1, 0, 2, 1, -2, 1), nrow = 3, byrow = TRUE) matrixb\_2 <- matrix(c(1, 1, 2, 2, 0, 1), nrow = 2, byrow = TRUE) result <- matrixb\_1 %\*% matrixb\_2 print(result)

# Part C

## The expression is not solvable. The number of columns in the first matrix must match the number of rows in the second matrix. The expression does not meet that need.

matrixc\_1 <- matrix(c(1, 0, 2, 1, -2, 1), nrow = 3, ncol = 2) matrixc\_2 <- matrix(c(1, -1, 0), nrow = 1, ncol = 3)

resultc <- matrixc\_1 %\*% matrixc\_2 print(resultc)

# Problem 3

# Part A

## This is solvalble. See below

a <- matrix(c(2, 1, 1, 2), nrow = 2, ncol = 2, byrow = TRUE) result <- round(a %\*% solve(a), 10) print(result)

# Part B

## This is not solvable. The determinate of the matrix is zero.

b <- matrix(c(2, 0, 1, 0), nrow = 2, ncol = 2, byrow = TRUE) result <- round(b %\*% solve(b), 10) print(result)

# Problem 4

# Import Data sets

women <- read.csv(“<https://raw.githubusercontent.com/EricBrownTTU/ISQS6350/main/womens_track.csv>”) men <- read.csv(“<https://raw.githubusercontent.com/EricBrownTTU/ISQS6350/main/mens_track.csv>”)

# Part A

cov(women[, 1:7]) cor(women[, 1:7])

# Part B

cov(men[,1:8]) cor(men[,1:8])

women men

# Part C

women\_five <- women[1:5, ] men\_five <- men[1:5, ]

common\_columns <- c(“m100”, “m200”, “m400”, “m800”, “m1500”, “m3000”, “marathon”)

women\_common <- women\_five[, common\_columns] men\_common <- men\_five[, common\_columns]

combined\_data <- rbind(women\_common, men\_common)

distance\_matrix <- dist(combined\_data[, -8], method = “euclidean”)

distance\_matrix <- round(distance\_matrix, 2)

print(distance\_matrix)

# Part D

The most similar entries are in row 4 and column 3, with a distance of 2.11. This is because these entries have the smallest euclidian distance. The most dissimilar entries are in row 7 and column , with a distance of 42.81. This is because these entries have the largest euclidean distance.

# Problem 5

# Part A

matrix\_5a <- matrix(c( 3.8778, 2.8110, 3.1480, 3.5062, 2.8110, 2.1210, 2.2669, 2.5690, 3.1480, 2.2669, 2.6550, 2.8341, 3.5062, 2.5690, 2.8341, 3.2352 ), nrow = 4, ncol = 4)

cor\_matrix <- cov2cor(matrix\_5a)

# Part B

cov\_1x2 <- matrix\_5a[1, 2] std\_dev\_1 <- sqrt(matrix\_5a[1, 1]) std\_dev\_2 <- sqrt(matrix\_5a[2, 2])

correlation\_1x2 <- cov\_1x2 / (std\_dev\_1 \* std\_dev\_2)

print(correlation\_1x2)

The correlation is 0.9801619

# Problem 6

# Part A

The jitter function is useful in this scenario because it may effectively demonstrate the correlation between two separate values originating from various variables, each of which contains numerous values inside the dataset.

grad <- read.csv(“<https://raw.githubusercontent.com/EricBrownTTU/ISQS6350/main/pgs.csv>”) grad

library(ggplot2) ggplot(data = grad, aes(x = jitter(FacTeaching), y = jitter(FacKnowledge))) + geom\_point() + labs(x = “FacTeaching”, y = “FacKnowledge”) + ggtitle(“Scatterplot of FacTeaching vs. FacKnowledge”)

# Part B

## New Data Frame

new\_data\_frame <- grad[, c(“FacTeaching”, “FacKnowledge”, “Housing”)]

new\_data\_frame

# Part C

na\_counts <- sapply(new\_data\_frame, function(x) sum(is.na(x)))

na\_counts

cleaned\_data <- na.omit(new\_data\_frame)

There are 17 NA values in FacTeaching, 16 NA values in FacKnowledge and 317 NA values in Housing

#Part C(i)

complete\_data <- cleaned\_data[complete.cases(cleaned\_data), ]

correlation\_matrix\_complete <- cor(complete\_data)

print(correlation\_matrix\_complete)

# Part C(ii)

correlation\_matrix\_available <- cor(new\_data\_frame, use = “complete.obs”)

print(correlation\_matrix\_available)

# Part C(iii)

correlation\_matrix\_ml <- cor(new\_data\_frame, use = “pairwise.complete.obs”)

print(correlation\_matrix\_ml)

The correlation matrix values derived from the entire and available case analyses are consistent, however,there is some deviation when compared to the results produced from the maximum likelihood estimate approach. Based on this observation, available-case analysis is preferred because it successfully manages missing values while providing results similar to those obtained after removing the missing-value observations.

# Problem 7

# load library

library(MVA)

# upload dataset

fish <- read.csv(“<https://raw.githubusercontent.com/EricBrownTTU/ISQS6350/main/fish.csv>”)

# subsets of the data

data\_subset <- fish[, c(“weight”, “length3”)] data\_subset\_2 <- fish[, c(“weight”, “hgtpct”)]

# Remove rows with missing values

data\_subset <- na.omit(data\_subset) data\_subset\_2 <- na.omit(data\_subset\_2)

# function to generate bivariate boxplots

bivariate\_boxplot <- function(data, x\_label, y\_label) { bvbox(data, xlab = x\_label, ylab = y\_label) }

# bivariate boxplots for weight vs. length3

bivariate\_boxplot(data\_subset, “Weight”, “Length3”)

# bivariate boxplots for weight vs. hgtpct

bivariate\_boxplot(data\_subset\_2, “Weight”, “Hgtpct”)

# original data and impute missing values with medians

fishn <- fish for (n in 1:ncol(fishn)) { fishn[is.na(fishn[, n]), n] <- median(fishn[, n], na.rm = TRUE) }

# correlations for the original data

cor\_original\_weight\_length3 <- cor(fishnlength3) cor\_original\_weight\_hgtpct <- cor(fishnhgtpct)

# remove outliers

outliers <- match(c(“100”, “101”, “102”, “103”, “104”, “153”, “155”, “157”), rownames(fishn)) data\_without\_outliers\_weight\_length3 <- fishn[-outliers, c(“weight”, “length3”)] cor\_without\_outliers\_weight\_length3 <- cor(data\_without\_outliers\_weight\_length3)

newoutliers <- match(c(“100”, “101”, “102”, “103”), rownames(fishn)) data\_without\_outliers\_weight\_hgtpct <- fishn[-newoutliers, c(“weight”, “hgtpct”)] cor\_without\_outliers\_weight\_hgtpct <- cor(data\_without\_outliers\_weight\_hgtpct)

# Problem 8

swiss <- read.csv(“<https://raw.githubusercontent.com/EricBrownTTU/ISQS6350/main/swiss.csv>”)

# a.

library(ggplot2)

# Calculation for kernel density estimates for top\_margin and diag\_length

density\_top\_margin <- density(swissdiag\_length, bw = 1)

# Plot kernel density estimates for top\_margin

ggplot(data = data.frame(x = density\_top\_marginy), aes(x = x, y = y)) + geom\_line() + labs(title = “Kernel Density Estimate of Top Margin”, x = “Length (mm)”, y = “Density”)

# Plot kernel density estimates for diag\_length

ggplot(data = data.frame(x = density\_diag\_lengthy), aes(x = x, y = y)) + geom\_line() + labs(title = “Kernel Density Estimate of Diagonal Length”, x = “Diagonal Length (mm)”, y = “Density”)

# b.

library(MASS) # Bivariate Gaussian Kernel Density Estimation kde\_bandwidth <- 1 # Use the same bandwidth as in part (a)

# The Estimate for bivariate density using a Gaussian kernel

kde\_data <- kde2d(swissdiag\_length, n = 100, h = c(kde\_bandwidth, kde\_bandwidth))

# contour plot

contour(kde\_data, main = “Bivariate Density - Contour Plot”, xlab = “Top Margin”, ylab = “Diagonal Length”)

# perspective plot

persp(kde\_data, main = “Bivariate Density - Perspective Plot”, xlab = “Top Margin”, ylab = “Diagonal Length”, zlab = “Density”)

# c.

#Scatterplot and Color Coding ggplot(swiss, aes(x = top\_margin, y = diag\_length, color = note)) + geom\_point() + labs(title = “Scatterplot of Top Margin vs. Diagonal Length”, x = “Top Margin”, y = “Diagonal Length”) + scale\_color\_manual(values = c(“genuine” = “blue”, “fake” = “red”)) + theme\_minimal()

# Problem 9

# Load package

data(BCG, package = “HSAUR2”)

# Exclude the year and the study number (columns 1 and 7)

bcg\_data <- BCG[, -c(1, 7)]

# a.

#Find the column-means vector column\_means <- colMeans(bcg\_data) cat(“Column Means:”) print(column\_means)

# b.

#Find the covariance matrix cov\_matrix <- cov(bcg\_data) cat(“Matrix:”) print(cov\_matrix)

# c.

#Find the Mahalanobis distances using the data and the values determined in parts (a) and (b) library(stats) mahalanobis\_distances <- mahalanobis(bcg\_data, center = column\_means, cov = cov\_matrix) cat(“Distances:”) print(mahalanobis\_distances)

# d.

#Sort the Mahalanobis distances from smallest to largest sorted\_indices <- order(mahalanobis\_distances) sorted\_distances <- mahalanobis\_distances[sorted\_indices] cat(“Mahalanobis Distances:”) print(sorted\_distances)

# e.

#Plot the sorted Mahalanobis distances vs the chi-square quantiles

df <- ncol(bcg\_data) chi\_sq\_quantiles <- qchisq(ppoints(length(sorted\_distances)), df) plot(chi\_sq\_quantiles, sorted\_distances, xlab = “Chi-Square Quantiles”, ylab = “Sorted Mahalanobis Distances”, main = “Q-Q Plot”) abline(0, 1, col = “red”)

# f.

#Interpretation The plot demonstrates that it is multivariate normal since the distance between the points is what is known as Mahalanobis distance.