A1 Assignment - Zino Meyer

1.1)

For this exercise, we work our way backwards from the the SVD results to the matrix A. We use the SVD equation $A = UDV^t$.

To focus on the first column, we will use the first column of the identity matrix i_1.

That way, we can compute $A_1 = UDV^t \cdot i_1$.

Starting from the right:

$$V^T \cdot e_1 = \begin{pmatrix} -1 & 0 \\ 0 & -1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 \\ 0 \end{pmatrix}$$

Next, we multiply by D:

$$D \cdot (V^T \cdot i_1) = \begin{pmatrix} 1.414214 & 0 \\ 0 & 1.414214 \end{pmatrix} \begin{pmatrix} -1 \\ 0 \end{pmatrix} = \begin{pmatrix} -1.414214 \\ 0 \end{pmatrix}$$

Finally, we calculate $U \cdot (D \cdot V^T \cdot i_1)$:

$$U \cdot (D \cdot V^T \cdot i_1) = \begin{pmatrix} -0.7071068 & -0.7071068 \\ -0.7071068 & 0.7071068 \end{pmatrix} \begin{pmatrix} -1.414214 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$$

Successfully recovered the first column: $A_1 = \begin{pmatrix} 1 \\ 1 \end{pmatrix}$

1.2)

```
# ------------------------
# Use R and the updated Darwin and Tahiti standardized SLP data to repro-
# duce the EOFs and PCs and to plot the EOF pattern maps and PC time series.

# Load data & keep years as row names
darwin_stand <- read.table("data/PSTANDdarwin.txt",
  header = FALSE, row.names = 1
)
years <- as.numeric(rownames(darwin_stand)) # save years for later
```

```
# Choose dec monthly data
darwin_dec <- t(darwin_stand[, 12])
colnames(darwin_dec) <- rownames(darwin_stand[, 12, drop = FALSE])

# Same for tahiti
tahiti_stand <- read.table("data/PSTANDtahiti.txt",
  header = FALSE, row.names = 1
)
tahiti_dec <- t(tahiti_stand[, 12])
colnames(tahiti_dec) <- rownames(tahiti_stand[, 12, drop = FALSE])

# Create a 2 by 65 space-time matrix from darwin and tahiti deceember data
da_ta <- rbind(darwin_dec, tahiti_dec)
dim(da_ta)
da_ta

# Calculate the SVD
da_ta_svd <- svd(da_ta)
U <- da_ta_svd$u
V <- t(da_ta_svd$v)

### Plot the EOF pattern maps
eof1 <- U[1, ]
eof2 <- U[2, ]

library(maps)
library(mapdata)

# Longitude and latitude for Darwin and Tahiti
locations <- data.frame(
  name = c("Darwin", "Tahiti"),
  lon = c(130.84, 210.58),
  lat = c(-12.46, -17.65)
)

# Initiate plot
plot.new()
par(mfrow = c(2, 1))
par(mar = c(0, 0, 0, 0)) # Zero space between (a) and (b)

# Mode 1 / EOF 1
map(database = "world2Hires", ylim = c(-70, 70), mar = c(0, 0, 0, 0))
grid(nx = 12, ny = 6)
points(locations$lon, locations$lat,
  pch = 19, col = c("blue", "red"), cex = 1.5
```

```
)
text(locations$lon, locations$lat,
  labels = paste0(locations$name, " ", round(eof1, 2)),
  col = "blue", pos = 1, cex = 1
)
axis(2,
  at = seq(-70, 70, 20),
  col.axis = "black", tck = -0.05, las = 2, line = -0.9, lwd = 0
)
axis(1,
  at = seq(0, 360, 60),
  col.axis = "black", tck = -0.05, las = 1, line = -0.9, lwd = 0
)
text(180, -50, "SLP Anomalies Darwin and Tahiti Mode 1",
  col = "purple", cex = 1.3
)
box()


# Mode 2 / EOF 2
map(database = "world2Hires", ylim = c(-70, 70), mar = c(0, 0, 0, 0))
grid(nx = 12, ny = 6)
points(locations$lon, locations$lat,
  pch = 19, col = c("blue", "red"), cex = 1.5
)
text(locations$lon, locations$lat,
  labels = paste0(locations$name, " ", round(eof2, 2)),
  col = "blue", pos = 1, cex = 1
)
axis(2,
  at = seq(-70, 70, 20),
  col.axis = "black", tck = -0.05, las = 2, line = -0.9, lwd = 0
)
axis(1,
  at = seq(0, 360, 60),
  col.axis = "black", tck = -0.05, las = 1, line = -0.9, lwd = 0
)
text(180, -50, "SLP Anomalies Darwin and Tahiti Mode 1",
  col = "purple", cex = 1.3
)
box()

dev.off()
```

```
### Plot the PC time series with matrix V
# png("images/pc-time-series-1-2.png", width = 800, height = 600)
plot(years, V[1, ],
  type = "l", col = "red",
  xlab = "Year", ylab = "PC", main = "PC Time Series"
)
lines(years, V[2, ], type = "l", col = "blue")
legend("topright", legend = c("PC1", "PC2"), col = c("red", "blue"), lty = 1)

dev.off()
```

1.8)

```
# ------------------------
# 1.8
# a)
precip_data <- read.csv("data/3447060.csv", header = TRUE)

# Create the pivot table with stations as rows, years as columns
station_year_df <- xtabs(PRCP ~ STATION + DATE, data = precip_data)

# Anomaly data X (with respect to mean), and Y columns for time in years
mean_precip <- rowMeans(station_year_df)
X <- sweep(station_year_df, 1, mean_precip)
Y <- ncol(X)

# covariance matrix formula: C = X * t(X) / Y
cov_x <- (X %*% t(X)) / Y
cov_x

# b)
inv_cov_x <- solve(cov_x)
inv_cov_x

# c)
eig_cov_x <- eigen(cov_x)
eig_cov_x$values
eig_cov_x$vectors

# d)
x_svd <- svd(X)
x_svd$u
```

```
x_svd$d
x_svd$v

# e)
eig_cov_x$values
x_svd$d
x_svd$d^2 / Y
# -> the eigenvalues of the covariance matrix are
# the squares of the singular values of the data matrix,
# divided by the number of columns

# f)
eig_cov_x$vectors
x_svd$u
# -> The eigenvectors of the covariance matrix are the same
# as the SVD spatial modes of X

# g) Plot the PC time series and describe their behavior.
V_t <- t(x_svd$v)
V_t

png("images/pc-time-series.png", width = 800, height = 600)
plot(1:Y, V_t[1, ], type = "l", col = "red",
    xlab = "Year", ylab = "PC", main = "PC Time Series")
lines(1:Y, V_t[2, ], type = "l", col = "blue")
lines(1:Y, V_t[3, ], type = "l", col = "green")
legend("topright", legend = c("1", "2", "3"),
    col = c("blue", "red", "green"), lty = 1)
dev.off()

# We see that lines one and three are very similar at the end (falling near
# year 4 and rising for year 5).
# Also, lines two and three are similar at the beginning (rising for year 1-2
# and falling after year 3).
# This describes / looks like an oscillation pattern, where all the lines
# rise and fall from -0.6 to 0.5. We just see an excerpt from this
# larger pattern.
# I would guess that one oscillation cycle is roughly 6-8 years long.
```

1.10)

Given: we need to balance the equation $C_3H_8 + xO_2 \rightarrow yCO_2 + zH_2O$

Solution:

Carbon: $1 \times 3 = y \Rightarrow y = 3$

Hydrogen: $1 \times 8 = 2z \Rightarrow z = 4$

Oxygen: $2x = 2y + z$

Substituting C and H into O:

⇒ $2x = 2 \times 3 + 4$

⇒ $2x = 10$

⇒ $x = 5$

Thus, the balanced equation is: $C_3H_8 + 5O_2 \rightarrow 3CO_2 + 4H_2O$

2.1)

```
# ------------------------
# 2.1 Write a computer code to
# a) Read the NOAAGlobalTemp data file, and
# b) Generate a 4 × 8 space-time data matrix for the December mean
# surface air temperature anomaly data of four grid boxes and eight years.

# a)
data_2_1 <- read.csv("data/NOAAGlobalT.csv", header = TRUE)
dim(data_2_1)

# b)
# Select San Diego, Berlin, Tokyo, Auckland
global_data <- data_2_1[c(1777, 2019, 1828, 755), ]

# Extract December data columns (every 12th column & without first 3 columns)
dec_cols <- seq(15, ncol(global_data), by = 12)

# Get december data only
dec_data <- global_data[, dec_cols]

# Extract years 2000-2008
data_2000_2008 <- dec_data[121:129]

# Set row and column names
row.names(data_2000_2008) <- c("San Diego", "Berlin", "Tokyo", "Auckland")
```

```
colnames(data_2000_2008) <- 2000:2008
data_2000_2008
```

2.2)

```
# ------------------------
# 2.2 Write a computer code to find the inverse of the following matrix.
matrix_2_2 <- matrix(
  c(
    1.7, -0.7, 1.3,
    -1.6, -1.4, 0.4,
    -1.5, -0.3, 0.6
  ),
  nrow = 3,
  byrow = TRUE
)

inverse_matrix <- solve(matrix_2_2)
print(inverse_matrix)
```

2.3)

```
# ---------------
# 2.3 Write a computer code to solve the following linear system of
# equations Ax = b, where A is a 3x3 matrix and b is a 3x1 vector:

A <- matrix(
  c(
    1, 2, 3,
    4, 5, 6,
    7, 8, 0
  ),
  nrow = 3, byrow = TRUE
)
b <- c(1, -1, 0)
x <- solve(A, b)
print(x)
```

2.4)

a)

To proof, I will express one row as a linear combination of the others explicitly. For the matrix

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \\ 7 & 8 & 9 \end{bmatrix}$$

the third row can be expressed as a combination of the first two:

$$a \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix} + b \begin{bmatrix} 4 \\ 5 \\ 6 \end{bmatrix} = \begin{bmatrix} 7 \\ 8 \\ 9 \end{bmatrix}$$

This equation has a solution for $a = -1$ and $b = 2$, showing that row 3 (vector 3) is a linear combination of row 1 & 2 (vectors 1 & 2), thus, the rows are linearly dependant

b)

Since row 3 can be expressed by row one and two, we reduce the system of equations by row three and end up with:

1. $x_1 + 2x_2 + 3x_3 = 0$
2. $4x_1 + 5x_2 + 6x_3 = 0$

Treating $x3$ as arbitraty value, we can rewrite the system like this:

1. $x_1 + 2x_2 = -3x_3$
2. $4x_1 + 5x_2 = -6x_3$

where x_3 is our arbitrary value, and x_1 and x_2 are variables.

c)

Solve the two equations for x_1 and x_2 and express them in terms of x_3 :

1. $x_1 + 2x_2 = -3x_3$
2. $4x_1 + 5x_2 = -6x_3$

Use elimination to solve for x_2:

$$4 \times 1. - 2.$$

⇒ $3x_2 = -6x_3$

⇒ $x_2 = -2x_3$

Substitude x_2 into $1.$:

$x_1 + 2(-2x_3) = -3x_3$

⇒ $x_1 - 4x_3 = -3x_3$

⇒ $x_1 = x_3$

So the solutions are:

$x_1 = x_3$

$x_2 = -2x_3$

$x_3 = x_3$

2.5)

Given: we need to balance the equation $2C_2H_6 + xO_2 \rightarrow yCO_2 + zH_2O$

Solution:

Carbon: $2 \times 2 = y \Rightarrow 4 = y$

Hydrogen: $2 \times 6 = 2z \Rightarrow 6 = z$

Oxygen: $2x = 2y + z$

Substituting C and H into the Oxygen equation:

⇒ $2x = 2(2 \times 2) + \frac{1}{2}(2 \times 6)$

⇒ $2x = 8 + 6$

⇒ $x = 7$

The balanced equation: $2C_2H_6 + 7O_2 \rightarrow 4CO_2 + 6H_2O$

2.7)

Given matrix & vector:

$$A = \begin{bmatrix} 0 & 4 \\ -2 & -7 \end{bmatrix}, u = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

a)

$$\mathbf{Au} = \begin{bmatrix} 0 & 4 \\ -2 & -7 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 0 \times 1 + 4 \times 1 \\ -2 \times 1 + (-7) \times 1 \end{bmatrix} = \begin{bmatrix} 4 \\ -9 \end{bmatrix}$$

Since $\mathbf{Au} \neq 4 \times \mathbf{u}$, where $4 \times \mathbf{u} = \begin{bmatrix} 4 \\ 4 \end{bmatrix}$, it shows that $\mathbf{u}$ is not an eigenvector of $\mathbf{A}$. The vector is not only scaled, but also changes its direction.

b)

```
# --------------
# 2.7 b)

A <- matrix(c(0, 4, -2, -7), nrow = 2, byrow = TRUE)

# eigenvalues & vectors:
eig <- eigen(A)
eigenvalues <- eig$values
eigenvectors <- eig$vectors

# Get unit eigenvectors by normalizing (divide by magnitude)
unit_eigenvectors <- apply(eigenvectors, 2, function(v) v / sqrt(sum(v^2)))
unit_eigenvectors
```