ST557: HOMEWORK 1

Brian Cervantes Alvarez
November 1, 2023

Question 1

```
# Set random seed for reproducibility
set.seed(503)

# Read HW1Q1 dataset in using base R
ds <- read.csv("HW1Q1.csv")
# Show first 5 rows
print(head(ds,50))
```

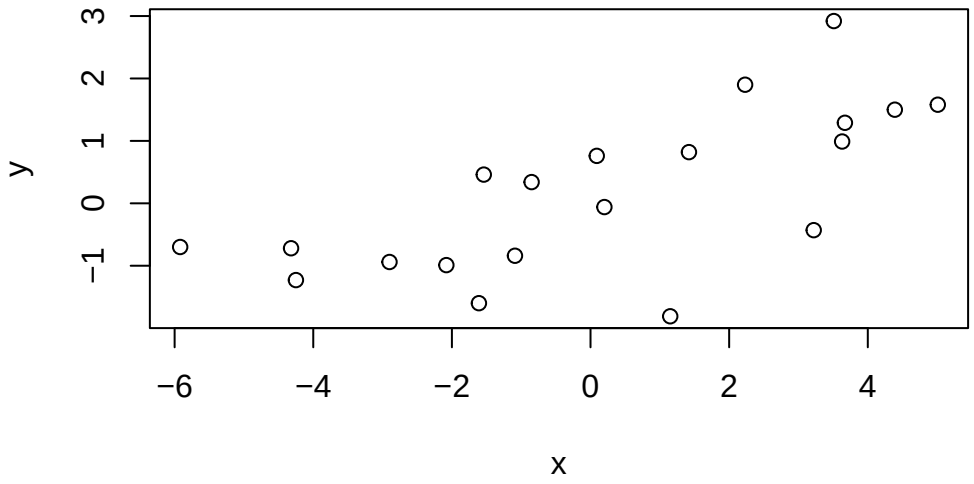
```
       X     Y
1  -1.54  0.46
2  -4.25 -1.23
3  -0.85  0.34
4  -2.90 -0.94
5  -1.09 -0.84
6  -5.92 -0.70
7   3.51  2.92
8   0.09  0.76
9  -2.08 -0.99
10  5.01  1.58
11  3.22 -0.43
12  3.67  1.29
13 -1.61 -1.60
14  2.23  1.90
15  0.20 -0.06
16  4.39  1.50
17  1.15 -1.81
18  3.63  0.99
19 -4.32 -0.72
20  1.42  0.82
```

Part A

Read this data into R using read.csv(). Create a 2-dimensional scatter plot of the 20 observations (use plot() function in R).

```r
# Get X and Y Components into vectors
x = ds$X
y = ds$Y

# Create scatter plot of Y ~ X
plot(x,y)
```



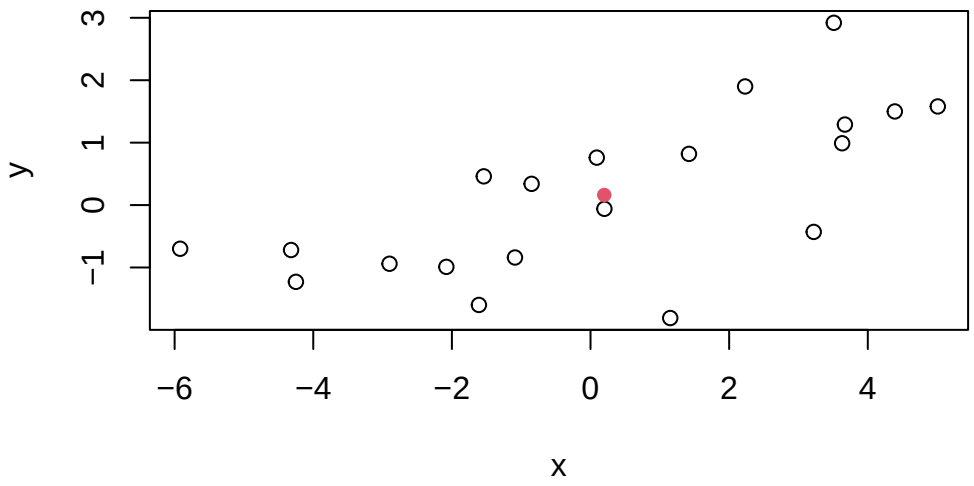
Part B

Find the sample mean vector, and add this point to the plot using points(). You can make this point a different color using the col= argument, or you can make it a different plotting character using the pch= argument. For example: > points(sampMean[1], sampMean[2], pch=16, col=2

```
plot(x,y)

sampMeanX = mean(x)
sampMeanY = mean(y)

points(sampMeanX, sampMeanY, pch=16, col=2)
```



Part C

Find the sample covariance matrix.

```
# Sample covariance matrix
sampCov <- cov(ds)

sampCov
```

```
          X        Y
X 10.140227 2.852078
Y  2.852078 1.668133
```

Part D

Find the eigendecomposition (spectral decomposition) of the sample covariance matrix using eigen().

```
sampEigenDecomp <- eigen(sampCov)

sampEigenDecomp
```

```
eigen() decomposition
$values
[1] 11.0108859  0.7974741

$vectors
           [,1]       [,2]
[1,] -0.9564274  0.2919702
[2,] -0.2919702 -0.9564274
```

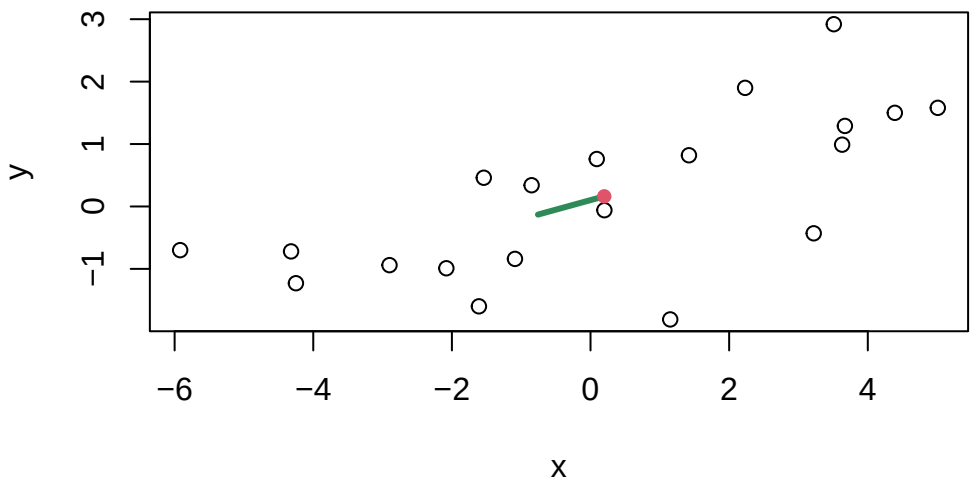
Part E

Add the eigenvector corresponding to the largest eigenvalue to the plot as a vector from the sample mean using `lines()`. Be careful here: if the first eigenvector is (v_1, v_2) and the sample mean vector is $(\bar{x}_1, \bar{x}_2)$, you want a line from $(\bar{x}_1, \bar{x}_2)$ to $(\bar{x}_1 + v_1, \bar{x}_2 + v_2)$."

```r
# Find the index of the largest eigenvalue
largestEigenvalue <- which.max(sampEigenDecomp$values)
# Extract the eigenvector corresponding to the largest eigenvalue
largestEigenvector <- sampEigenDecomp$vectors[, largestEigenvalue]

# Calculate the endpoints of the line segment
sampMeanX <- mean(x)
sampMeanY <- mean(y)
xend <- sampMeanX + largestEigenvector[1]
yend <- sampMeanY + largestEigenvector[2]

plot(x,y)
# Add the line segment to the plot
lines(c(sampMeanX, xend), c(sampMeanY, yend), col = "seagreen4", lwd = 3)
points(sampMeanX, sampMeanY, pch=16, col=2)
```



Discussion for Part E

Question: Describe how the direction of this eigenvector relates to the cloud of data points

The eigenvector follows the trend of the data points, which are showing a positive increasing trend, and it culminates precisely at the location of the sample means for X and Y.

Question 2

Part A

```r
ds <- read.csv("CubitData.csv")
head(ds,5)
```

```
    height    cubit
1 70.98437 18.60823
2 70.82176 18.49283
3 70.62555 19.25116
4 71.31924 17.34156
5 71.35977 18.51334
```

```r
# Calculate the sample mean vector
sampleMeanVec <- colMeans(ds)
sampleMeanVec
```

```
  height    cubit
67.08137 18.07067
```

Part B

```
# Calculate the sample covariance matrix!
sampleCovMatrix <- cov(ds)
sampleCovMatrix
```

```
          height     cubit
height 5.604262 1.4548363
cubit  1.454836 0.8796708
```

Part C

```r
# Calculate the eigendecomposition using the covariance matrix, sampleCovMatrix
sampleEigenDecomp <- eigen(sampleCovMatrix)
sampleEigenDecomp
```

```
eigen() decomposition
$values
[1] 6.0163114 0.4676216

$vectors
           [,1]       [,2]
[1,] -0.9621535  0.2725080
[2,] -0.2725080 -0.9621535
```

Part D

```r
# Find the index of the largest eigenvalue
largestEigenvalue <- which.max(sampleEigenDecomp$values)
# Extract the eigenvector corresponding to the largest eigenvalue
largestEigenvector <- sampleEigenDecomp$vectors[, largestEigenvalue]
largestEigenvector
```

```
[1] -0.9621535 -0.2725080
```

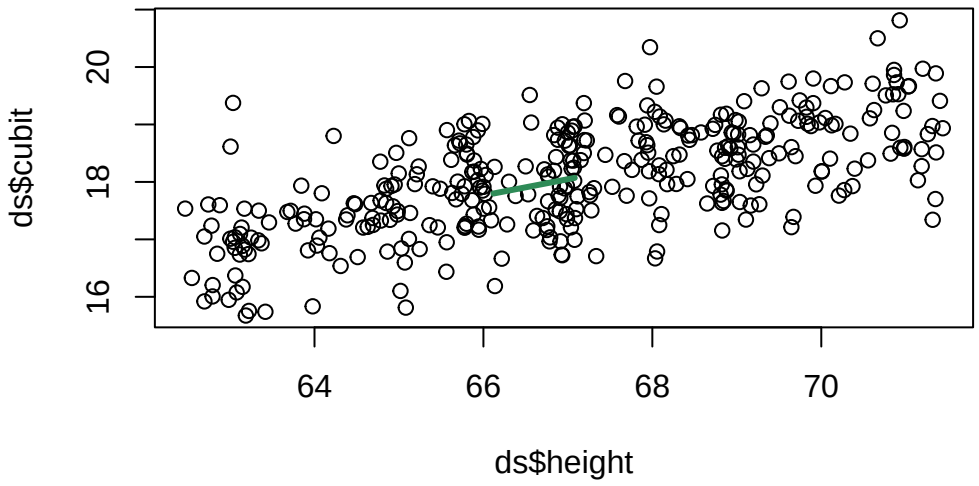
```r
# Create the scatter plot
plot(ds$height, ds$cubit)

# Calculate each sample mean
meanCubit <- mean(ds$cubit)
meanHeight <- mean(ds$height)

# Use Part D!

xend <- meanHeight + largestEigenvector[1]
yend <- meanCubit + largestEigenvector[2]


# Plot the eigenvector as a line from the mean point
lines(c(meanHeight, xend), c(meanCubit, yend), col = "seagreen4", lwd = 3)
```



Discussion for Part E

Question: Describe how the direction of this eigenvector relates to the cloud of data points

The eigenvector for the height and cubit follows the trend of the data points, which are showing a positive increasing trend. While not exactly the same, it's similar to problem 1.

Question 3

Part A

```r
A <- matrix(c(5.125, 3.875, 2.125, -1.125, 0,
              3.875, 5.125, -1.125, 2.125, 0,
              2.125, -1.125, 5.125, 3.875, 0,
              -1.125, 2.125, 3.875, 5.125, 0,
              0, 0, 0, 0, -3),
            nrow = 5,
            byrow = TRUE)

# Calculate the eigendecomposition
eigenDecomposition <- eigen(A)
eigenDecomposition
```

```
eigen() decomposition
$values
[1] 10.0  8.0  4.5 -2.0 -3.0

$vectors
     [,1] [,2] [,3] [,4] [,5]
[1,]  0.5  0.5 -0.5  0.5    0
[2,]  0.5  0.5  0.5 -0.5    0
[3,]  0.5 -0.5 -0.5 -0.5    0
[4,]  0.5 -0.5  0.5  0.5    0
[5,]  0.0  0.0  0.0  0.0    1
```

Part B

```r
# Grab the eigenvalues from the eigen decomposition
eigenvalues <- eigenDecomposition$values
# Use the function "all" to check all eigenvalues to see if they're positive definite
isPositiveDefinite <- all(eigenvalues > 0)
isPositiveDefinite
```

```
[1] FALSE
```

```r
# False! Now we need to find a vector x...

# Get all eigenvector(s) that are negative
negVec <- eigenDecomposition$vectors[, which(eigenvalues < 0)]

# We need to normalize the eigenvector(s) using the equation
vecNorm <- negVec / sqrt(sum(negVec^2))
vecNorm
```

```
           [,1]      [,2]
[1,]  0.3535534 0.0000000
[2,] -0.3535534 0.0000000
[3,] -0.3535534 0.0000000
[4,]  0.3535534 0.0000000
[5,]  0.0000000 0.7071068
```

```r
result <- t(vecNorm) %*% A %*% vecNorm
result
```

```
     [,1] [,2]
[1,]   -1  0.0
[2,]    0 -1.5
```

We end of getting a vector $x = \begin{bmatrix} 0.3535534 \\ -0.3535534 \\ -0.3535534 \\ 0.3535534 \\ 0 \end{bmatrix}$ for which $x^T Ax < 0$, confirming that A is not positive definite.

Part C

The matrix-vector multiplication Ax, where $x = 4v_1 + 2v_5$, can be expressed symbolically as:

$$Ax = 4\lambda_1 v_1 + 2\lambda_5 v_5$$

In this expression, λ_1 and λ_5 are the eigenvalues corresponding to v_1 and v_5, respectively. If we want to calculate it, we could plug in the eigenvalues and solve.

Question 4

Part A

```r
# Read the data from the CSV file
ds <- read.csv("IrisData.csv")
names(ds)
```

```
[1] "Sepal.Length" "Sepal.Width"  "Petal.Length" "Petal.Width"  "Type"
```

```r
# Calculate the sample mean vector for all variables
meanVec <- colMeans(ds[,])
meanVec
```

```
Sepal.Length  Sepal.Width Petal.Length  Petal.Width         Type
    5.843333     3.057333     3.758000     1.199333     2.000000
```

```r
print(ds$Type)
```

```
  [1] 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
 [38] 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2
 [75] 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 2 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[112] 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3 3
[149] 3 3
```

Part B

```
# Calculate the sample mean vector for each species
species1MeanVec <- colMeans(ds[ds$type == 1, 1:4])
species2MeanVec <- colMeans(ds[ds$type == 2, 1:4])
species3MeanVec <- colMeans(ds[ds$type == 3, 1:4])
```

Part C

```r
# Calculate the sample correlation matrix for all variables
corMatrix <- cor(ds[,])
corMatrix
```

```
             Sepal.Length Sepal.Width Petal.Length Petal.Width       Type
Sepal.Length    1.0000000  -0.1175698    0.8717538   0.8179411  0.7825612
Sepal.Width    -0.1175698   1.0000000   -0.4284401  -0.3661259 -0.4266576
Petal.Length    0.8717538  -0.4284401    1.0000000   0.9628654  0.9490347
Petal.Width     0.8179411  -0.3661259    0.9628654   1.0000000  0.9565473
Type            0.7825612  -0.4266576    0.9490347   0.9565473  1.0000000
```

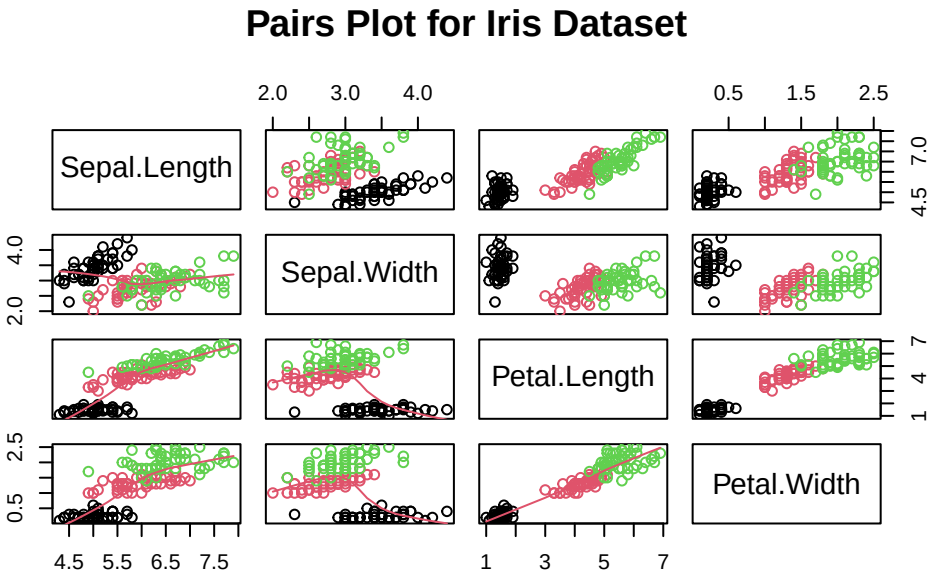
Part D

```r
# Calculate individual correlation matrices for each species
species1CorMatrix <- cor(ds[ds$type == 1, 1:4])
species2CorMatrix <- cor(ds[ds$type == 2, 1:4])
species3CorMatrix <- cor(ds[ds$type == 3, 1:4])
```

Part E

```
library(ggplot2)

pairs(ds[, 1:4],
      main = "Pairs Plot for Iris Dataset",
      pch = 21, col = as.numeric(ds$Type),
      labels = colnames(ds)[1:4],
      lower.panel = panel.smooth,
)
```



You can totally spot some noticeable differences among these flowers. When it comes to telling them apart, Petal Width seems to be the way to go, thanks to their clear clustering that lets you easily identify the species. And if you look at Petal Length and Petal Width, it's pretty apparent that they both follow a nice linear pattern in their clustering, making it even easier to tell them apart based on these features.

Question 5

Part A

To show that B is a symmetric matrix, we need to demonstrate that B equals its transpose, i.e., B = B^T. We have:

$$B = A^T A$$

Taking the transpose of B:

$$B^T = (A^T A)^T = A^T (A^T)^T = A^T A = B$$

Thus, B is a symmetric matrix.

Part B

To show that B is a positive semi-definite matrix, we need to prove that for any vector x in $\mathbb{R}^p$, $x^T Bx \geq 0$. Let's calculate this expression:

$$x^T Bx = x^T(A^T A)x = (x^T A^T)(Ax) = (Ax)^T(Ax) = \|Ax\|^2$$

Since the square of the Euclidean norm (length) of any vector is non-negative ($\|v\|^2 \geq 0$ for any vector v), we have $x^T Bx \geq 0$. Therefore, B is a positive semi-definite matrix.

Part C

The sample covariance matrix S can be expressed as:

$$S = \frac{1}{n-1}(X - \bar{X})^T(X - \bar{X})$$

Where $\bar{X}$ is an $(n \times p)$ matrix with n identical rows equal to the sample mean vector $(\bar{X}_1, \bar{X}_2, \ldots, \bar{X}_p)$. To show that S is positive semi-definite, we need to prove that for any vector x in $\mathbb{R}^p$, $x^T S x \geq 0$.

Let $y = (X - \bar{X})x$, which is a linear combination of the columns of $X - \bar{X}$. Now, the expression $x^T S x$ can be written as:

$$x^T S x = \frac{1}{n-1} y^T y$$

Since y is a linear combination of the columns of $X - \bar{X}$, y is a vector in $\mathbb{R}^n$. The squared norm of any vector in $\mathbb{R}^n$ is non-negative. Therefore, $\frac{1}{n-1} y^T y \geq 0$, and as a result, $x^T S x \geq 0$.

This shows that the sample covariance matrix S is positive semi-definite.