# Sheet #1

## Data Matrix

| ID | a1 | a2 | a3 | a4 |
|---|---|---|---|---|
| 1 | 10 | 60 | 10 | 90 |
| 2 | 20 | 50 | 40 | 70 |
| 3 | 30 | 50 | 30 | 40 |
| 4 | 20 | 50 | 20 | 60 |
| 5 | 10 | 60 | 30 | 10 |

DATA MATRIX **D**

1. Given the Data Matrix on the right answer the following questions.
    a. What is number of dimensions?
    b. What are the types of the attributes?
    c. What is the distance between x1 and x3?
    d. What is the length of x2?
    e. What is the cos(angle) between x2 and x4?
    f. Do we need attribute scaling?
    g. Compute the attribute scaled data matrix after scaling each attribute linearly between 0 and 1.
    h. Repeat parts c, d, e on the scaled data matrix in part (g)
2. Given the Data Matrix on the right submit your python code and its output that will do the following.
    a. Compute the norm of each instance. (5x1).
    b. Compute the Cosine similarity matrix (5x5) matrix.
    c. Compute the Euclidean Distance matrix of the instances (5x5)

## Principal Component Analysis

3. Given Data matrix above. Consider a1, a2 and a4 only.
    a. Write down the new data matrix **D3** (5x3).
    b. Plot the data using 3d scatter plots.
    c. Compute the mean vector (3x1).
    d. Compute centered data matrix **Z** by subtracting mean vector from the Data Matrix. (5x3).
    e. Compute Covariance matrix **COV** (3x3).
    f. Use python solvers to find eigenvalues (Diagonal 3x3 matrix) and eigenvectors (3x3) matrix. **Take care of the eigenvalues order**.
    g. Verify $U^T \wedge U$= COV.

h. Compute the explained variance by the eigenvector corresponding to the largest eigenvalue. Do you think one eigenvector is good enough?
i. Compute the projection matrix **P** to go to 2-dimensions. Consider the top two eigenvectors of matrix **U** according to eigenvalues. (3x2)
j. Project the instances into a 2-Dimension space. $\mathbf{x}_{\upsilon} = \mathbf{P}^{T}\mathbf{x}$
k. Plot the resulting Data matrix **D2** using scatter plots.

4. We have 4 data points. The following is their data matrix, on which we want to apply the PCA.

| **X1** | **X2** |
|---|---|
| 6 | -4 |
| -3 | 5 |
| -2 | 6 |
| 7 | -3 |

a. Compute the mean and the covariance of the given data matrix.
b. Knowing that the unit eigenvectors of the covariance matrix are (1/√2, 1/√2) with eigenvalue = 2 and (-1/√2, 1/√2) with eigenvalue = 162, find the first principal component of the PCA.

5. Given the data below, answer the following questions:

a. Compute 3x3 Covariance matrix of the 5 tuples dataset we have.
b. If the trace of the covariance matrix is the sum of the eigenvalues of the matrix.
   i. Compute the three eigenvalues of the covariance matrix if
   $$\frac{\lambda_a}{\lambda_b} = 0.505 \ and \ \frac{\lambda_b}{\lambda_c} = 0.647, where \ \lambda_a < \lambda_b < \lambda_c$$
   ii. Determine the explained variance using only $\lambda_b, \lambda_c$

| | X1 | X2 | X3 |
|---|---|---|---|
| x1 | 0.5 | 4.5 | 2.5 |
| x2 | 2.2 | 1.5 | 0.1 |
| x3 | 3.9 | 3.5 | 1.1 |
| x4 | 2.1 | 1.9 | 4.9 |
| x5 | 0.5 | 3.2 | 1.2 |

## Hints

- numpy.linalg.eigh(A) A is a matrix. Computes eigenvectors and eigenvalues of symmetric matrix A
- numpy.dot(A,B) A, B are matrices/vectors. Computes dot product
- numpy.mean(A, axis=0) A is matrix, axis =0 will average over the columns
- numpy.diag(A) converts vector A into a diagonal matrix.
- numpy.vstack((A,B)) expand matrices A,B into one wider matrix → number of rows of A and B must match.
- numpy.transpose(A) will compute the transpose of a matrix/vector A.

## Notes

- Make sure you did everything on your own.
- Marks are put on trial and effort not on correct answer, so cheating will be severely penalized.
- Deliver the solution in PDF format on the email:

  patternssp2021@gmail.com with subject [Sheet1][ID]