UNIVERSITY OF OULU

FACULTY OF INFORMATION TECHNOLOGY AND ELECTRICAL ENGINEERING

MATLAB Homework

Statistical Signal Processing I
8th of October 2023

Athmajan Vivekananthan
avivekan23@student.oulu.fi
2305573

Hathurusingha Anjalee Anuththara De Silva
hdesilva23@student.oulu.fi
2303245

CONTENTS

1 Exercise-I

1.1

Create a 5x5 random matrix A with entries drawn from a Gaussian distribution N(0,1).
A = randn([5,5])

The **randn** command generates a random scalar from a standard normal distrinution. The input arguments to the **randn** function here is passed as an array giving the dimensions of the required matrix (A)[1].

1.2

Compute the singular value decomposition (SVD) of A.

$$[U, S, V] = svd(A)$$

The function svd(A) decomposes matrix such that A = U*S*V'[2].
By capturing the **Left Singular Vectors matrix** (U), **Right Singular Vectors matrix** (V) and the **Singular Value matrix** (S) containing the singular values of A as the return variables - U,S and V as an array, the Singular Value decomposition can be obtained[2].

1.3

Reconstruct A using the SVD components, and name it A_{svd}.

$$A_{svd} = U * S * V'$$

The approximation of matrix A by Singular Value Decomposition can be obtained by right multiplying the U,S, and V^T in that particular order.

1.4

Verify the reconstruction quality by computing the Frobenius norm $||A - A_{svd}||_F^2$.

$$n = norm(A - Asvd, "fro"))$$

The Frobenius norm can be calculated by the function **norm** and passing the first argument as the matrix that we want to find the norm for and the second argument with the keyword **fro** standing for Frobenius norm.[3]
For the instance that was generated for A and A_{svd}, the value for Frobenius norm obtained was $n = 3.824 \times 10^{-15}$. This value is very close to zero and can be used to verify that the matrix **A** and matrix $\mathbf{A_{svd}}$ are super close to each other.

2 Exercise-II

Consider the following probability density function of the continuous uniform distribution:

$$f(x) = \frac{1}{b-a}, if a \le x \le b \tag{2.1}$$

In order to find the lower bound and upper bound of the uniform distribution in terms of μ and σ^2 the following calculation is done.

$$\begin{aligned} E[x] &= \int_a^b x \times \frac{1}{b-a} dx \\ &= \frac{1}{b-a} \times \frac{x^2}{2}|_a^b \\ &= \frac{b^2-a^2}{2} \times \frac{1}{b-a} \\ &= \frac{a+b}{2} \end{aligned} \tag{2.2}$$

$$\begin{aligned} E[x^2] &= \int_a^b x^2 \times \frac{1}{b-a} dx \\ &= \frac{1}{b-a} \times \frac{x^3}{3}|_a^b \\ &= \frac{1}{3} \times (b^3 - a^3) \frac{1}{b-a} \\ &= \frac{a^2+ab+b^2}{3} \end{aligned} \tag{2.3}$$

$$\begin{aligned} \sigma^2 &= E[x^2] - (E[x])^2 \\ &= \frac{a^2+ab+b^2}{3} - \frac{(a+b)^2}{4} \\ &= \frac{4 \times (a^2+ab+b^2) - 3 \times (a^2+2ab+b^2)}{12} \\ &= \frac{a^2-2ab+b^2}{12} \\ &= \frac{(a-b)^2}{12} \end{aligned} \tag{2.4}$$

From (2.2) and (2.4)

$$\begin{aligned} a &= -1 \\ b &= 5 \end{aligned} \tag{2.5}$$

The validity of the calculation can be found by the following Matlab command :

$$[M, V] = unifstat(-1, 5)$$

Where M and V are the mean and variances of a uniform distribution given the lower an upper boundaries. The outputs show **M = 2** and **V = 3** which verifies the calculation of a and b to be correct[4].

2.1

Generate 1000 random samples, where each sample is drawn from the uniform distribution with mean $\mu = 2$ *and variance* $\sigma^2 = 3$.

```
r = -1 + (6).*rand(1000,1)
```

The Matlab function **rand** can be used to generate a random scalar drawn from a uniformly distributed probability model. Pushing in the argument of (1000,1) produces a column vector of 1000 elements with the starting lower boundary **(a = -1)** and the length of the uniformity **(b - a = 6)**[4].

2.2

Plot a histogram of the generated samples.

```
histogram(r,30)
```

The histogram of the obtained uniformly distributed 1000 samples by the function **histogram**[5].

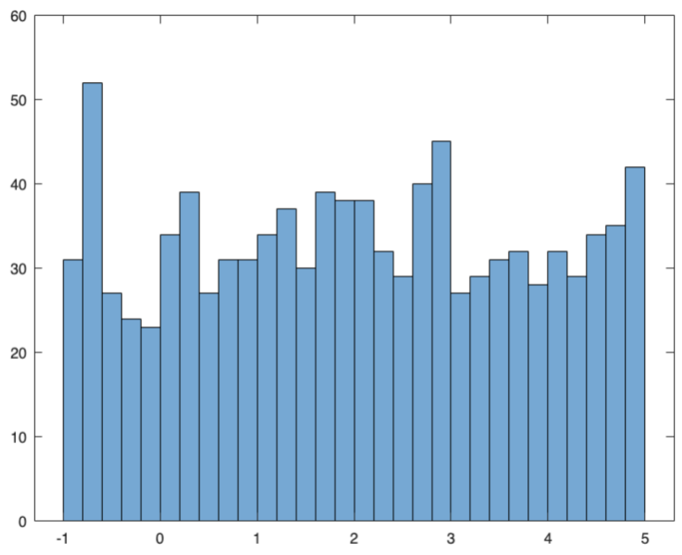


Figure 2.1. Histogram of 1000 samples of U [-1,5]

3 Exercise-III

*Suppose you have a 2D matrix **D** with dimensions 500 × 2.*

3.1

*Generate the first column of **D**, namely **x**, that contains 500 samples drawn from a Gaussian distribution with mean $\mu = 0.5$ and variance $\sigma^2 = 0.5$*

x = normrnd(0.5,sqrt(.5),[500,1])

The matlab function **normrnd** generates a random number from the standard normal distribution with the desired mean and standard deviation as arguments. Therefore the square root of the variance has been fed in to the function in terms of the standard deviation. The last argument in the function is the size of the vector desired as the output from the function[6].

3.2

*Generate the second column of **D**, namely **y**, that contains 500 samples drawn from a Gaussian distribution with mean $\mu = 0$ and variance $\sigma^2 = 2$*

y = normrnd(0,sqrt(2),[500,1])

Similar to Section 3.1 we can use the Matlab function **normrnd** to generate Gaussian samples[6].

3.3

*Compute the sample mean and the sample variance of **z = x + y**, and compare them with the theoretical mean and the theoretical variance of **z**.*

The new Random Variable **z** is derived as the addition of **x** and **y**. Therefore the samples derived from the section 3.1 and section 3.2 can be added to derive a sample for **z**.

$$\begin{aligned} z &= x + y \\ v &= var(z) \\ m &= mean(z) \end{aligned}$$

The mean and variance of the new random variable **z** can be obtained using simple term by term addition of **x** and **y** and then by using the **var** and **mean** Matlab functions to obtain the variance and mean respectively.

The results obtained for the generated samples in Matlab are as follows:

$$v = 2.2282 \tag{3.1}$$

$$m = 0.3968 \tag{3.2}$$

In order to compare the above obtained results on (3.1) and (3.2) with the theoretical variance and mean of z, the following calculation needs to be done:

$$\begin{aligned} \mathbf{X} &= N(\mu_x, \sigma_x^2) \\ \mathbf{Y} &= N(\mu_y, \sigma_y^2) \end{aligned} \tag{3.3}$$

By substituting the values for mean and variance:

$$\begin{aligned} \mathbf{X} &= N(0.5, 0.5) \\ \mathbf{Y} &= N(0, 2) \end{aligned} \tag{3.4}$$

Since **X** and **Y** are generated independent to each other, they are independent Gaussian random variables where their PDFs can be defined as below:

$$f_X(x) = \frac{1}{\sqrt{2\pi}\sigma_x} \times \exp(\frac{-(x-\mu_x)^2}{2\sigma_x^2}) \tag{3.5}$$

$$= \frac{1}{\sqrt{\pi}} \times \exp(-(x-0.5)^2)$$

$$f_Y(y) = \frac{1}{\sqrt{2\pi}\sigma_y} \times \exp(\frac{-(y-\mu_y)^2}{2\sigma_y^2}) \tag{3.6}$$

$$= \frac{1}{2\sqrt{\pi}} \times \frac{\exp(-y^2)}{4} \tag{3.7}$$

$$\tag{3.8}$$

Since **Z** = **X** + **Y**, **Y** can be written as **Y** = **Z** - **X**.
Therefore writing PDF of **Z** using (3.7) in terms of **X** and **Z** gives:

$$f_Z(z) = \int_{-\infty}^{+\infty} f_X(x) f_Y(z-x) dx \tag{3.9}$$

$$= \int_{-\infty}^{+\infty} \frac{1}{\sqrt{\pi}} \times \exp(-(x-0.5)^2) \times \frac{1}{2\sqrt{\pi}} \times \frac{\exp(-(z-x)^2)}{4} \tag{3.10}$$

$$= \frac{1}{\sqrt{5\pi}} \times \exp(-\frac{(z-0.5)^2}{5}) \tag{3.11}$$

$$\tag{3.12}$$

(3.11) can be written to be able to see the Gaussian parameters explicitly as below:

$$f_Z(z) = \frac{1}{\sqrt{2\pi \times (2.5)}} \times \exp(-\frac{(z-0.5)^2}{2 \times 2.5}) \qquad (3.13)$$

Therefore from (3.13) we can see that $\mu_z = 0.5$ and $\sigma_z^2 = 2.5$

Parameter	Calculated Value	Theoretical Value
μ_z	0.3968	0.5
σ_z^2	2.2282	2.5

Table 3.1. Comparison of theoretical and calculated values of **Z**

4 Exercise-IV

4.1

Create a 2×2 covariance matrix Σ with non-zero off-diagonal elements.

A covariance matrix is always both symmetric and positive semi- definite. Since the off diagonal elements of Σ are fixed to be non zero, Σ the covariance matrix becomes symmetric and positive definite.

$$E = [1, 0.5; 0.5, 1] \tag{4.1}$$

By defining a square matrix with prime numbers as elements and multiplying by its transpose we can make sure to obtain a symmetric positive definite matrix that would have the characteristics of a covariance matrix.

4.2

Generate a white Gaussian random vector w with dimension 2×1000*, where each column has a zero mean and an identity covariance matrix.*

```
w = mvnrnd([0 0],eye(2),1000)’
```

By using **mvnrnd** function of Matlab we can obtain a white Gaussian samples with zero mean as the first argument and identity matrix as the second argument whereas 1000 has been passed as the third argument. This would make the covariance matrix of w to be an Identity matrix[7].

4.3

Perform the Cholesky decomposition on Σ and obtain its lower triangular matrix L.

```
L = chol(E)’
```

The **chol** function on Matlab with the argument **’lower’** the Lower Triangular matrix can be found which would satisfy the E = L’*L. [8]

4.4

Use the matrix L to map w to a new Gaussian random vector x such that $x = L \times w$*.*

```
x = L*w
```

4.5

Compute the sample covariance matrix of x and verify that the sample covariance matrix is approximately equal to Σ.

$covsample = cov(x')$
$n = norm(E - covsample, "fro")$

The Frobenius norm of the difference matrix of E and covX is saved on the variable n. For the selected covariance matrix E, the value n takes for this experiment is n = n = 0.0732. Which is quite accurate to conclude that both E and covX are approximately equal[9].

5 REFERENCES

[1] Inc. T.M. (2023) Normally distributed random numbers. Technical report, The MathWorks Inc., Natick, Massachusetts, United States, https://www.mathworks.com/help/matlab/ref/randn.html.

[2] Inc. T.M. (2023) Singular value decomposition. Technical report, The MathWorks Inc., Natick, Massachusetts, United States, https://www.mathworks.com/help/matlab/ref/double.svd.html.

[3] Inc. T.M. (2023) Vector and matrix norms. Technical report, The MathWorks Inc., Natick, Massachusetts, United States, https://www.mathworks.com/help/matlab/ref/norm.html.

[4] Inc. T.M. (2023) Uniformly distributed random numbers. Technical report, The MathWorks Inc., Natick, Massachusetts, United States, https://www.mathworks.com/help/matlab/ref/rand.html.

[5] Inc. T.M. (2023) Histogram plot. Technical report, The MathWorks Inc., Natick, Massachusetts, United States, https://www.mathworks.com/help/matlab/ref/matlab.graphics.chart.primitive.histogram.html.

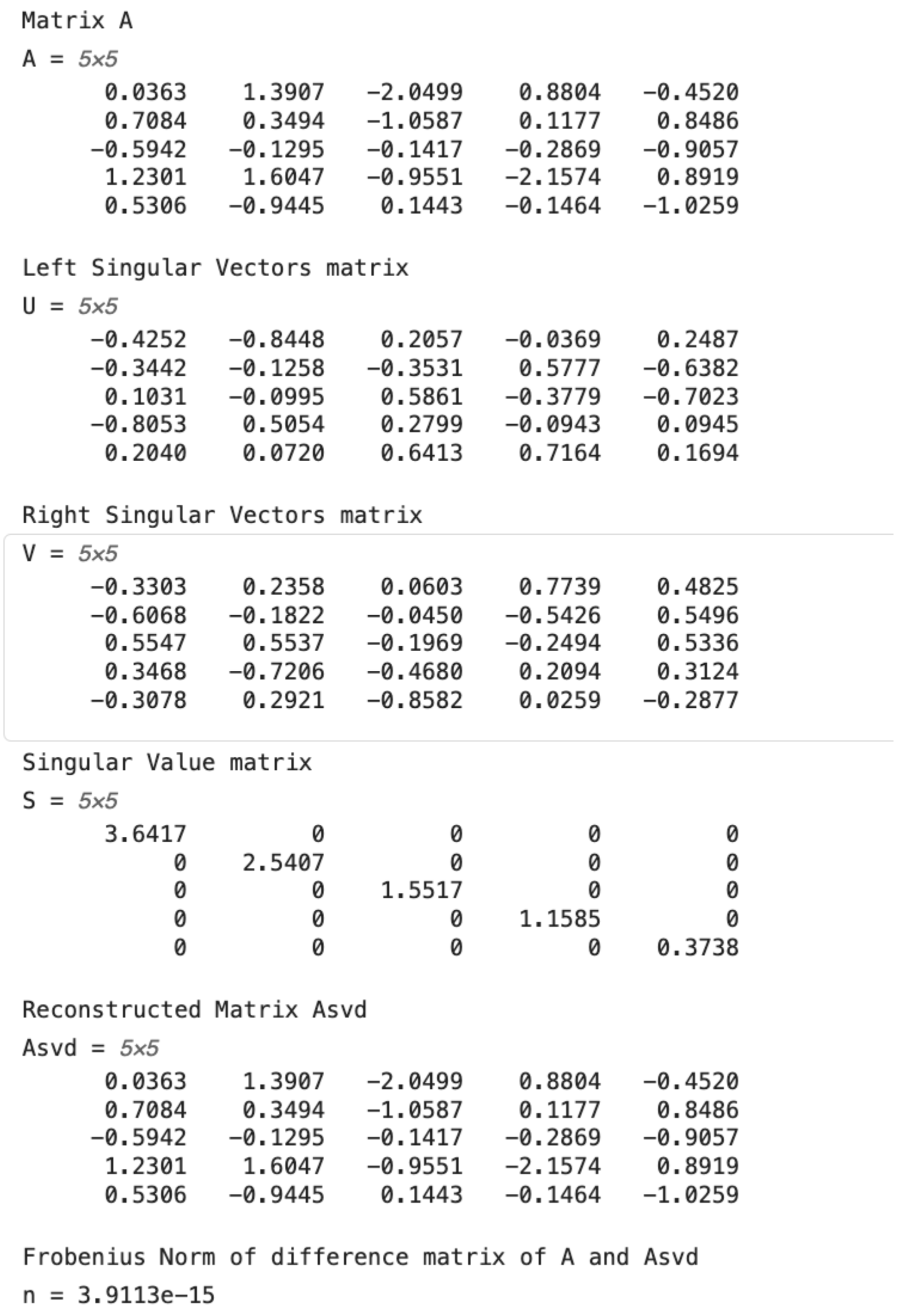
[6] Inc. T.M. (2023) Normal random numbers. Technical report, The MathWorks Inc., Natick, Massachusetts, United States, https://www.mathworks.com/help/stats/normrnd.html.

[7] Inc. T.M. (2023) Multivariate normal random numbers. Technical report, The MathWorks Inc., Natick, Massachusetts, United States, https://www.mathworks.com/help/stats/mvnrnd.html.

[8] Inc. T.M. (2023) Cholesky factorization. Technical report, The MathWorks Inc., Natick, Massachusetts, United States, https://www.mathworks.com/help/matlab/ref/chol.html.

[9] Inc. T.M. (2023) Covariancecollapse. Technical report, The MathWorks Inc., Natick, Massachusetts, United States, https://www.mathworks.com/help/matlab/ref/cov.html.

6 APPENDICES



```
Matrix A
A = 5x5
    0.0363    1.3907   -2.0499    0.8804   -0.4520
    0.7084    0.3494   -1.0587    0.1177    0.8486
   -0.5942   -0.1295   -0.1417   -0.2869   -0.9057
    1.2301    1.6047   -0.9551   -2.1574    0.8919
    0.5306   -0.9445    0.1443   -0.1464   -1.0259

Left Singular Vectors matrix
U = 5x5
   -0.4252   -0.8448    0.2057   -0.0369    0.2487
   -0.3442   -0.1258   -0.3531    0.5777   -0.6382
    0.1031   -0.0995    0.5861   -0.3779   -0.7023
   -0.8053    0.5054    0.2799   -0.0943    0.0945
    0.2040    0.0720    0.6413    0.7164    0.1694

Right Singular Vectors matrix
V = 5x5
   -0.3303    0.2358    0.0603    0.7739    0.4825
   -0.6068   -0.1822   -0.0450   -0.5426    0.5496
    0.5547    0.5537   -0.1969   -0.2494    0.5336
    0.3468   -0.7206   -0.4680    0.2094    0.3124
   -0.3078    0.2921   -0.8582    0.0259   -0.2877

Singular Value matrix
S = 5x5
    3.6417         0         0         0         0
         0    2.5407         0         0         0
         0         0    1.5517         0         0
         0         0         0    1.1585         0
         0         0         0         0    0.3738

Reconstructed Matrix Asvd
Asvd = 5x5
    0.0363    1.3907   -2.0499    0.8804   -0.4520
    0.7084    0.3494   -1.0587    0.1177    0.8486
   -0.5942   -0.1295   -0.1417   -0.2869   -0.9057
    1.2301    1.6047   -0.9551   -2.1574    0.8919
    0.5306   -0.9445    0.1443   -0.1464   -1.0259

Frobenius Norm of difference matrix of A and Asvd
n = 3.9113e-15
```

Figure 6.1. Matlab output for Exercise 1

```
r = 1000×1
      1.2806
      2.4726
      3.2000
      4.8732
      4.5035
      1.6879
      4.6902
      4.0740
      1.8497
      3.5895
        ⋮
```

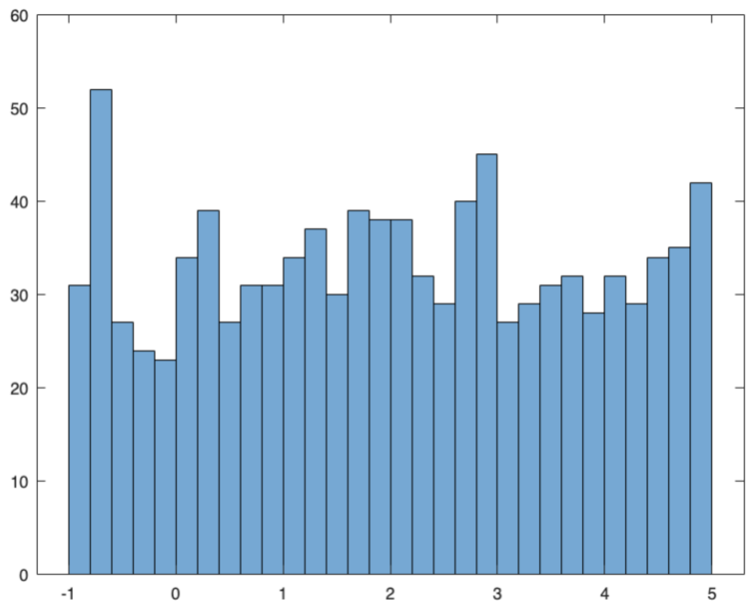
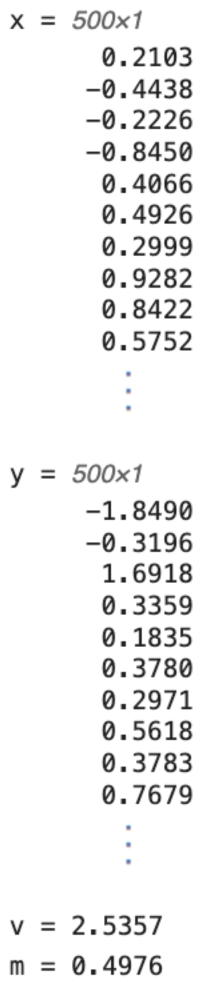


Figure 6.2. Matlab output for Exercise 2

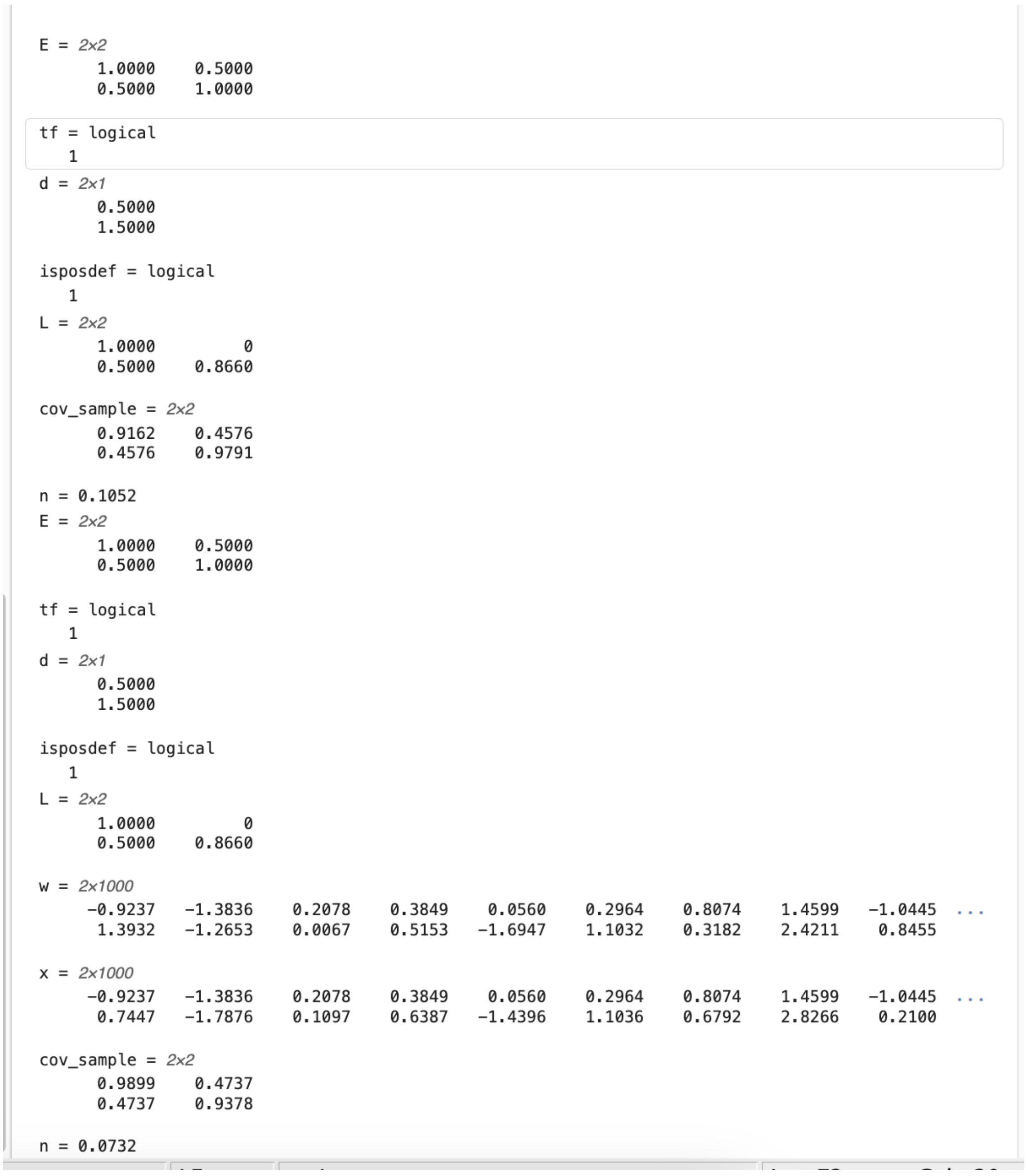


```
x = 500×1
      0.2103
     -0.4438
     -0.2226
     -0.8450
      0.4066
      0.4926
      0.2999
      0.9282
      0.8422
      0.5752
        ⋮

y = 500×1
     -1.8490
     -0.3196
      1.6918
      0.3359
      0.1835
      0.3780
      0.2971
      0.5618
      0.3783
      0.7679
        ⋮

v = 2.5357
m = 0.4976
```

Figure 6.3. Matlab output for Exercise 3



```
E = 2x2
      1.0000    0.5000
      0.5000    1.0000

tf = logical
   1
d = 2x1
      0.5000
      1.5000

isposdef = logical
   1
L = 2x2
      1.0000         0
      0.5000    0.8660

cov_sample = 2x2
      0.9162    0.4576
      0.4576    0.9791

n = 0.1052
E = 2x2
      1.0000    0.5000
      0.5000    1.0000

tf = logical
   1
d = 2x1
      0.5000
      1.5000

isposdef = logical
   1
L = 2x2
      1.0000         0
      0.5000    0.8660

w = 2x1000
     -0.9237   -1.3836    0.2078    0.3849    0.0560    0.2964    0.8074    1.4599   -1.0445  ...
      1.3932   -1.2653    0.0067    0.5153   -1.6947    1.1032    0.3182    2.4211    0.8455

x = 2x1000
     -0.9237   -1.3836    0.2078    0.3849    0.0560    0.2964    0.8074    1.4599   -1.0445  ...
      0.7447   -1.7876    0.1097    0.6387   -1.4396    1.1036    0.6792    2.8266    0.2100

cov_sample = 2x2
      0.9899    0.4737
      0.4737    0.9378

n = 0.0732
```

Figure 6.4. Matlab output for Exercise 4