Devin Trejo

ECE 3522: Stochastic Processes in Signals and Systems

Department of Electrical and Computer Engineering, Temple University, Philadelphia, PA 19122

I. PROBLEM STATEMENT

Today we introduce multivariable sample spaces. We will learn more about covariance or how closely two sets of data are correlated. Using MatLab we will be able to construct some random sets of data and observe the covariance. Specifically we will utilize MatLab's 3D visualization tools to observe the PDFs of our random vectors. Plotting in 3D will allow us to see the covariance and other general characteristics of our two random variables.

First we will start with generating two random arrays which are independently generated uniformly between zero and one. The task has us noting the general shape and meaning behind our 3D PDF or surf plot.

Second we will tell MatLab to construct our two random variables in accordance to four different covariance matrices (seen in Figure 2). Using these covariances and also a mean of [6 6] for both randomly generate variables, we are able to construct distributions which are not all normal. To observe the covariance we create a support region or plane which intersects the distribution as seen in Figure 1. We will see for covariances matrices which are not the identity matrix the variances of our random variables is not equal in all directions. Note how the diagonal corresponds to the variance of the values and the off diagonal tells us the correlation.

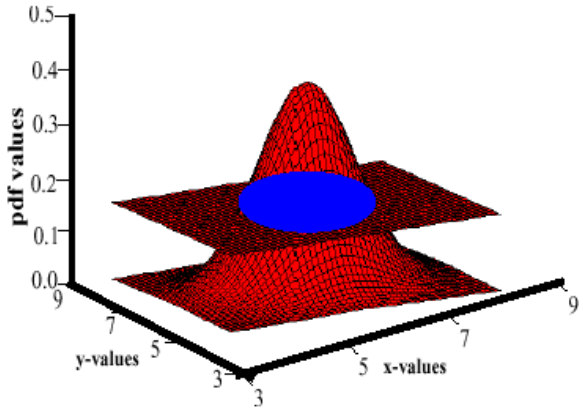

Figure 1: Support region.

Source:<ECE 8527: Lecture 2 ppt>

$$Cov(X,Y) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$$

$$Cov(X,Y) = \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix}$$

$$Cov(X,Y) = \begin{bmatrix} 1 & 0.5 \\ 0.5 & 1 \end{bmatrix}$$

$$Cov(X,Y) = \begin{bmatrix} 5 & 0.5 \\ 0.5 & 2 \end{bmatrix}$$

Figure 2: Covariance matrices for Part 2.

II. APPROACH AND RESULTS

A. Part 1

We begin with our first task of creating two randomly generated uniform variables between 0 and 1. We can use MatLab's random number generator to complete the task. To find the histogram of the 2 variables we use MatLab's function 'hist3'. In the previous assignment we emphasized and proved how the number of samples you take can influence whether your data reaches your expected output. Therefore we generate 100,000 vectors and sum all the results so to guarantee our output produces accurate results. Below we plot our overall histogram:

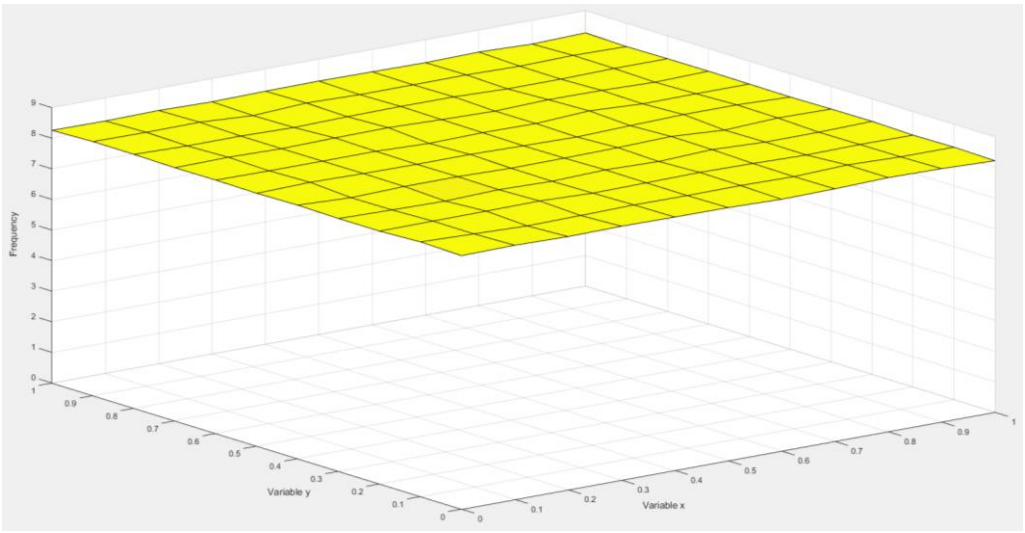

Figure 3: Uniform distribution of two randomly generated variables with binSize = 0.1.

Figure 4 showcases our resulting plot which matches our expectations. Since we a generating a 100,000 random variables between zero and one we observe how each combination has equal frequency of occurring. In 3D space this will correspond to a flat plane. With such a large sample size our results are not biased in any way. The above plot shows a 3D histogram with a bin size of 0.1 with a flat plane histogram. If we decrease the binSize to 0.01 we will observe a less uniform appearing plot as seen in Figure 5. However, both plots demonstrate the same set of data. With the smaller binSize we observe how in realty not even 100,000 points will produce completely uniform results. Between the two plots however we see the same characteristics where each bin has approximately the same frequency of occurring thus a uniform distribution.

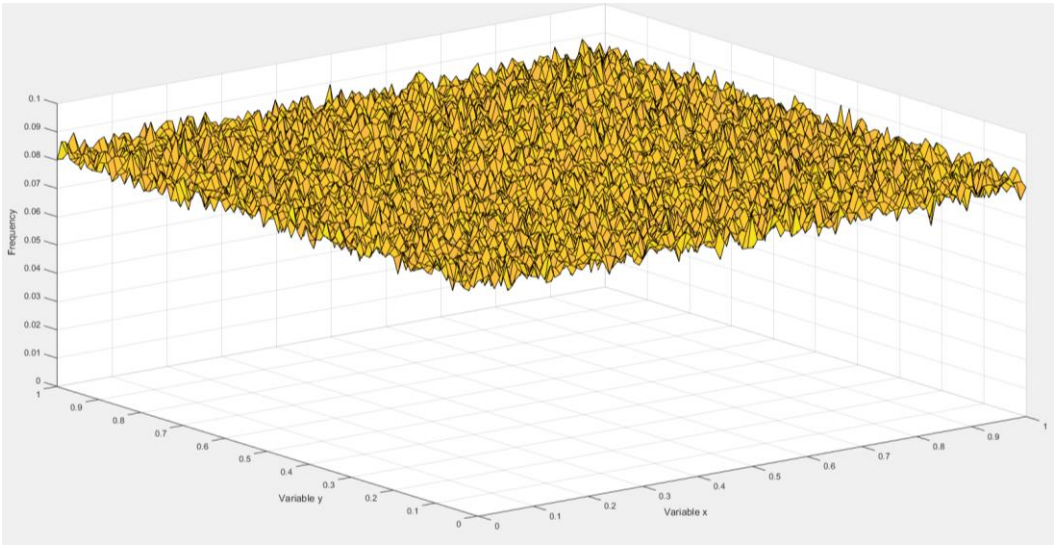

Figure 5: Uniform distribution of two randomly generated variables with binSize = 0.01.

B. Part 2

Next we move onto changing our covariance matrices so that are data is not normal. First we take the covariance matrix that matches the identity and thus should produce a normal 3D PDF. For all these following results we again take 100,000 sample vectors.

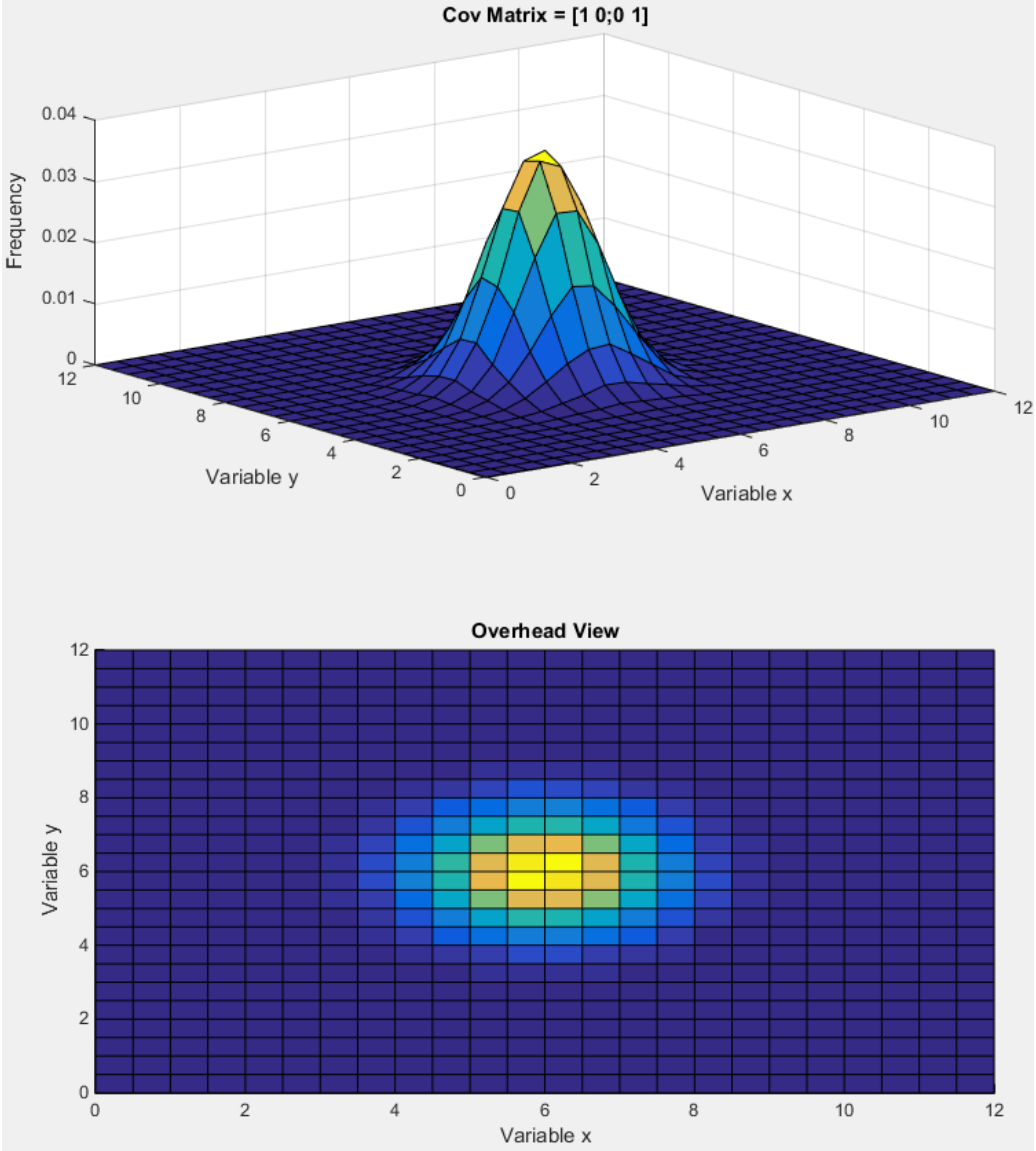

Figure 6: Multivariable PDF with identity covariance matrix (3D plot and support region)

Our first result is very normal where we have equal variance in all directions. It appears that plot there is correlation between X and Y. The off-diagonal of the covariance matrix states otherwise. From the plot we observe how the values centered one bin away from the mean occur more frequently. Since both X and Y are built off the same variance values it appears that the two are correlated but they are not. Both X and Y were generated separately using MatLab's random number generator. An ideal case can be seen below in Figure 7.

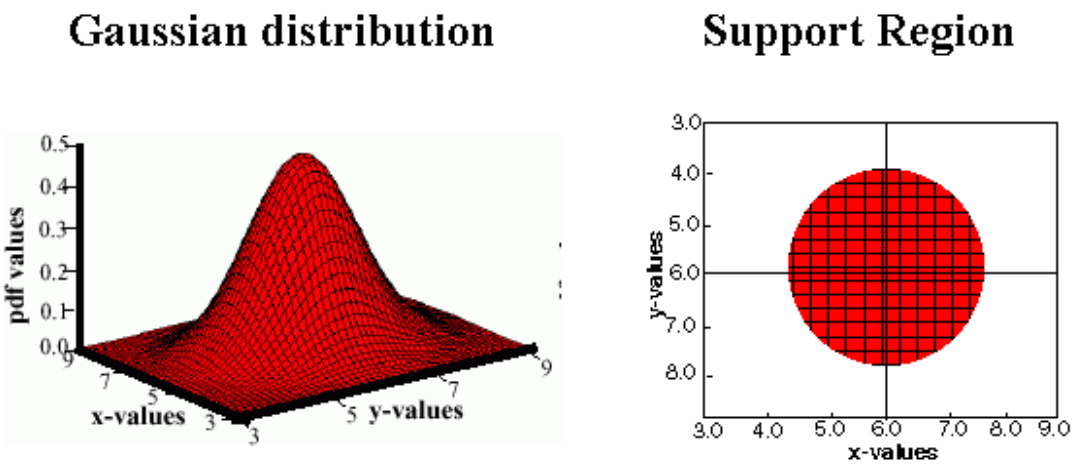

Figure 7 Expected Results given CovMatrix = [1 0; 0 1]

Source:<ECE 8527: Lecture 2 ppt>

We will continue running the experiment across a multitude of difference covariance matrices laid out in Figure 2. Note that as we change our covariance matrix our resulting PDF is skewed in the directions indicated by the matrix.

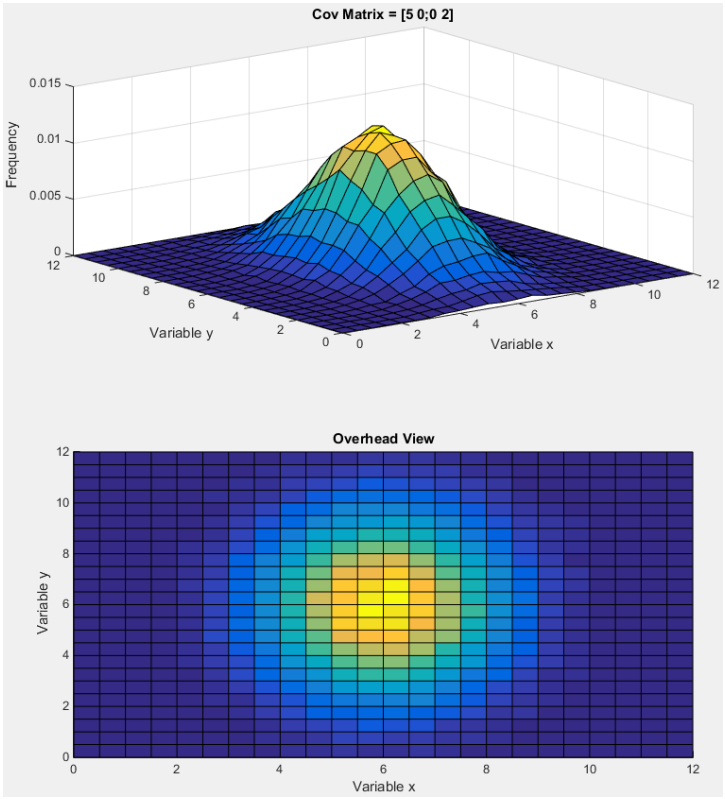

Figure 8: Multivariable PDF with CovMatrix = [5 0; 0 2] (3D plot and support region)

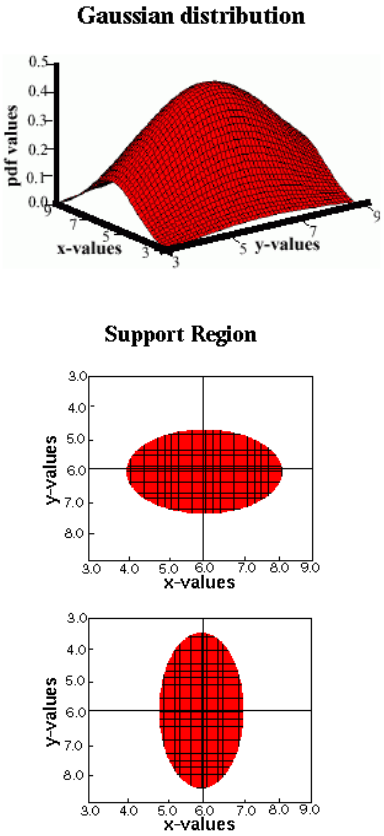

Figure 9: Expected Results given CovMatrix = [5 0; 0 2]

Source:<ECE 8527: Lecture 2 ppt>

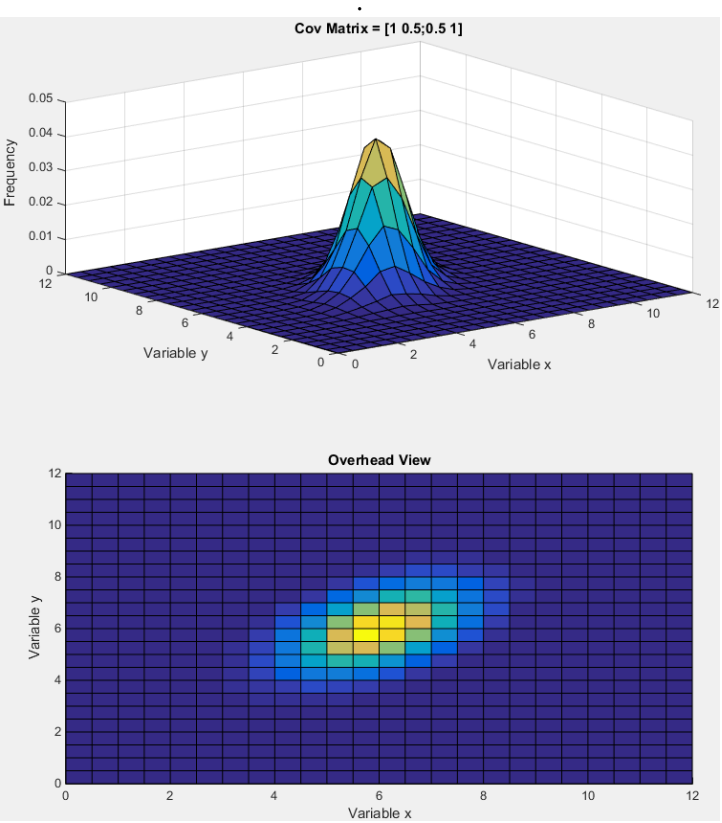

Figure 10: Multivariable PDF with CovMatrix = [1 0.5; 0.5 1] (3D plot and support region)

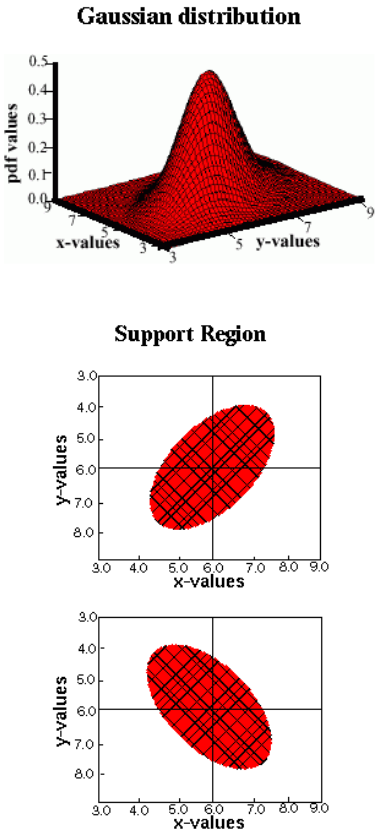

Figure 11: Expected Results given CovMatrix = [1 0.5; 0.5 1]

Source:<ECE 8527: Lecture 2 ppt>

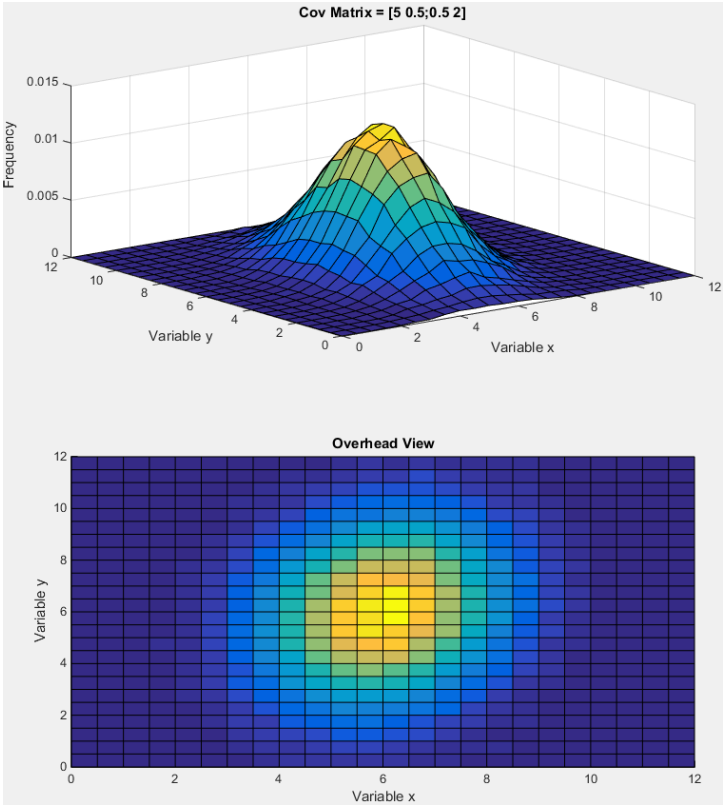

Figure 12: Multivariable PDF with CovMatrix = [5 0.5; 0.5 2] (3D plot and support region)

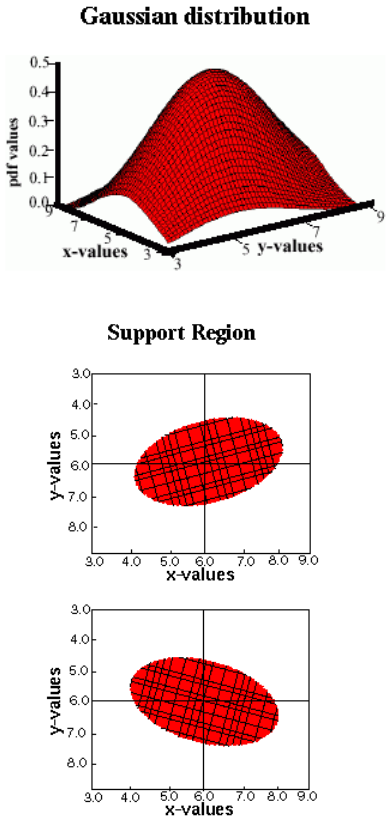

Figure 13: Expected Results given CovMatrix = [5 0.5; 0.5 2]

Source:<ECE 8527: Lecture 2 ppt>

From the three different plots above we can observe the effects of our covariance matrices. In the first example our matrix is $Cov(X,Y) = \begin{bmatrix} 5 & 0 \\ 0 & 2 \end{bmatrix}$. In this example there is correlation between X & Y since the diagonal is zero. The variance of X and Y are different where the variance for X values is 5 while for Y its only 2. Thus in the plot are X values take on a more spread of values compared to Y values which produces an ellipse support region.

The next example introduce correlation between X and Y. The resulting PDF will produce a support region where the ellipse is skewed. The example also holds the variance among the values to be 1. Comparing to our last case where our correlation is apparent again but the variance of X and Y are different. Since the variance is higher in the X sample our support region is shifted off the X-values axis but not as much as in Figure 10.

III. MATLAB CODE

```
clear; clc; close all

numSam = 10^4;
vecL = 1000;
binSize = 0.1;

% Lower and Upper bound of random Number Generation
lRange = 0;
uRange = 1;

% Histogram Bounds
bounds = [lRange:binSize:uRange];
numBins = length(bounds);

Edges = {bounds, bounds};
```

1 To begin we setup some parameters that will define the characteristics of our random data set. The 'numSam' is the amount of random two dimension random arrays we will create. Having a larger number of samples will allow us to reach a more uniform result as we observed in the previous class assignment. In the next MatLab statement we define 'vectL' which will allow us to set the length of each random independent vector. The 'binSize' is self-explanatory as defining the length of each bin our histogram will divide amongst. 'lRange' & 'uRange' define the lower and upper limits of our random number generator. Finally, we need to define an cell array 'Edges' which will allow us to pass the edges of each bin into our hist3 function which we will later use to find the histogram of our two dimensional array.

```
for N=1:numSam
    x(:,:,N) = lRange +(uRange-lRange)*rand(vecL,2);
    xHist(:,:,N) = hist3(x(:,:,N),'Edges', Edges);
end

% Find histogram for the entire set of N samples
sumxHist = sum(xHist,3);

%Normalize the histogram
sumxHist = sumxHist/(numSam);

%Plot 3D histogram
figure()
surf(bounds, bounds, sumxHist)
xlabel('Variable x');
ylabel('Variable y');
zlabel('Frequency');
xlim([0 1])
ylim([0 1])
```

2 First we create a 'numSam' of random vectors and compute the histogram each time. Using our N counter we keep track of each random vector we create as well as the histogram for each two dimensional random vector. After computing the histogram for 'numSam' vectors using 'hist3' we find the combined histogram of each by summing them up in the third dimension or the dimension we used to store each individual histogram. This newly created overall histogram is now normalized to the 'numSam' we computed. Lastly we plot our results in three dimensions using 'surf'. The 'bounds' parameter we defined previously is used to define the x and y axis of our 3-D plot.

```
clear; clc; close all

binSize = .5;
vecL = 100000;

% Lower and Upper bound of random Number Generation
lRange = 0;
uRange = 12;

% Histogram Bounds
bounds = [lRange:binSize:uRange];
numBins = length(bounds);

Edges = {bounds, bounds};

mu = [6,6];
coV(:,:,1) = [1 0; 0 1];
coV(:,:,2) = [5 0; 0 2];
coV(:,:,3) = [1 0.5; 0.5 1];
coV(:,:,4) = [5 0.5; 0.5 2];
```

3 Moving onto Part 2 of the assignment we start with defining our function parameters. Given our sample mean is a constant 6 we define our range to be ±6 from the mean. A 'binSize' of 0.5 will be adequate in obtaining quantifiable results and a 'vecL' of 100,000 will give us more points so our plots are smooth. At this point we also define our given covariance matrices and mean mu which we will use to find our random sample arrays.

```
for N = 1:4
    x(:,:,N) = mvnrnd(mu,coV(:,:,N), vecL);
    xHist(:,:,N) = hist3(x(:,:,N),'Edges',Edges);
end

%Normalize
xHist = xHist/vecL;
```

4 Using the 'mvnrnd' we are able to construct a multivariable random array given a set mean and covariance matrix. As is similar to the previous part we use 'hist3' to find our multivariable histogram. After we process each covariance matrix case, we break from our loop and normalize our results.

```
%Plot 3D histogram
figure()
subplot(2,1,1)
surf(bounds, bounds, xHist(:,:,1))
title(['Cov Matrix = ' mat2str(coV(:,:,1))]);
xlabel('Variable x');
ylabel('Variable y');
zlabel('Frequency');
subplot(2,1,2)
surf(bounds, bounds, xHist(:,:,1))
title('Overhead View');
xlabel('Variable x');
ylabel('Variable y');
zlabel('Frequency');
view(2)

figure()
subplot(2,1,1)
surf(bounds, bounds, xHist(:,:,2))
title(['Cov Matrix = ' mat2str(coV(:,:,2))]);
xlabel('Variable x');
ylabel('Variable y');
zlabel('Frequency');
subplot(2,1,2)
surf(bounds, bounds, xHist(:,:,2))
title('Overhead View');
xlabel('Variable x');
ylabel('Variable y');
zlabel('Frequency');
view(2)


figure()
subplot(2,1,1)
surf(bounds, bounds, xHist(:,:,3))
title(['Cov Matrix = ' mat2str(coV(:,:,3))]);
xlabel('Variable x');
ylabel('Variable y');
zlabel('Frequency');
subplot(2,1,2)
surf(bounds, bounds, xHist(:,:,3))
title('Overhead View');
xlabel('Variable x');
ylabel('Variable y');
zlabel('Frequency');
view(2)


figure()
subplot(2,1,1)
surf(bounds, bounds, xHist(:,:,4))
title(['Cov Matrix = ' mat2str(coV(:,:,4))]);
xlabel('Variable x');
ylabel('Variable y');
zlabel('Frequency');
subplot(2,1,2)
surf(bounds, bounds, xHist(:,:,4))
title('Overhead View');
xlabel('Variable x');
ylabel('Variable y');
zlabel('Frequency');
view(2)
```

5 Lastly we plot our individual histograms for each covariance case. To better see the cross sectional support region we create a subplot and re-plot the histogram using a forced a 2-dimension overhead view using the ‘view(2)’ function.

IV. CONCLUSIONS

The purpose of the assignment is to better understand and visualize multivariable datasets. To begin we focused on showing the PDF of a uniform distribution. What we developed was a flat plane. The 3D surf plot does a good job in demonstrating our PDF for both our random variables. We can clearly see how every combination of X and Y has the same frequency of occurring within our test case.

We also proved our understanding of the covariance matrix. Knowing that the diagonal corresponds to the variance of our variables and the off diagonal tells us the correlation between them, we can analyze their corresponding PDFs. An identity covariance matrix is a model of a perfectly evenly independent distributed multivariable array. The support region is a circle showing the even distribution in all directions. As we change the variance of our covariance matrix then our X and Y sample change accordingly. In the second example we change the variance of X values to 5 and Y to two. The PDF is an elongated ellipse where our X values have more spread of values. Note however there is no correlation between our X and Y values. In our third example we introduce a correlation of 0.5 between X and Y. The slight correlation introduces a skewness to our PDF. In the third example the variance of X and Y are equal so our PDF is skewed so our PDF is 45 degrees off the horizontal. On the contrary in example four our variance along the X values is higher but our X and Y are still correlated. The resulting PDF is also skewed but not as much of the horizontal X values axis.