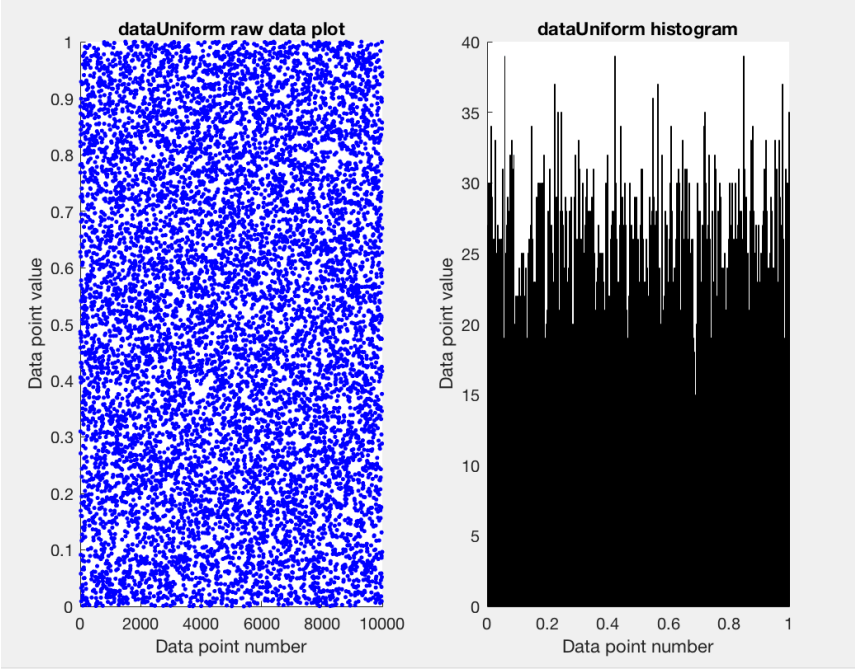
Student Number: 10450559

P1.1 1D and 2D distributions

1 The results are of a uniform probability distribution of random samples to make up a 1xn matrix. This data is taken from Matlab using the rand function which is used for uniform data. The 1xn matrix is plotted on both the histogram and the plot diagram in a uniform fashion which produces the results. The plot is very random and not correlated or focused on any particular area. The number of samples determine the size of the matrix/ the number of values to be plotted and the bins determines the number of intervals for the data in order for them to be organised and plotted onto the histogram. An increase of samples will only full the graph more with more random plotted points.

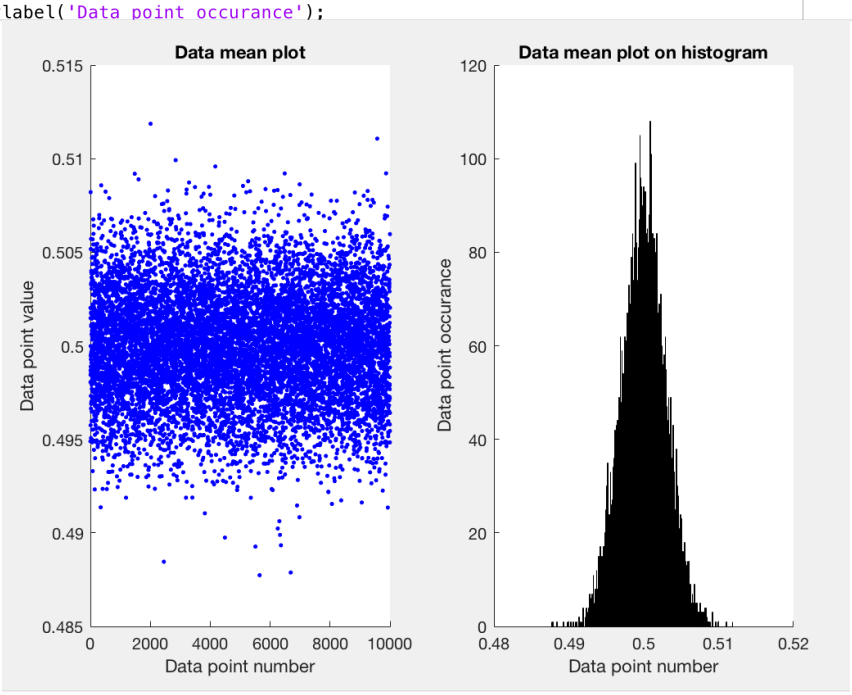
```
%Number of datapoints
samples = 10000;

%Create a Matrix of 1 X samples (1xn) from a uniform distribution
data = rand(1,samples);

%Specify the number of bins
nbins = 400;

%Print the matrix size
message = sprintf('dataUniform Matrix size %d x %d', size(data,1), size(data,2));
fprintf(message);

%Plot data on figure position 1
figure
subplot(1,2,1);
hold on
h = plot(data,'b.');
set(h,'linewidth',3);
title('dataUniform raw data plot');
xlabel('Data point number');
ylabel('Data point value');

%Plot data for the histogram on figure position 2
subplot(1,2,2);
hold on
histogram(data,nbins);
title('dataUniform histogram');
xlabel('Data point number');
ylabel('Data point value');
```



Student Number: 10450559

2 This shows the central limit theorem using a uniform distribution of samples similar to that of before. In this case a matrix of nxn is created from a random sample of the uniform distribution data. The 2nd dimension of the nxn matrix is used to calculate the average. The 2nd dimension represents the columns of the matrix, therefore when the columns of the matrix have an average applied across them each column uses the mean function through the entire matrix. This mean data was plotted on both a scatter plot and a histogram. From this we can see that the data is as if it is a normal distribution in both the plots cases. This is also a show of the central limit theorem. As before the increase sample size causes more plots to be plotted and an increase in bins causes more histogram bars to be plotted causing the plots to look even more like a normally distributed data set.

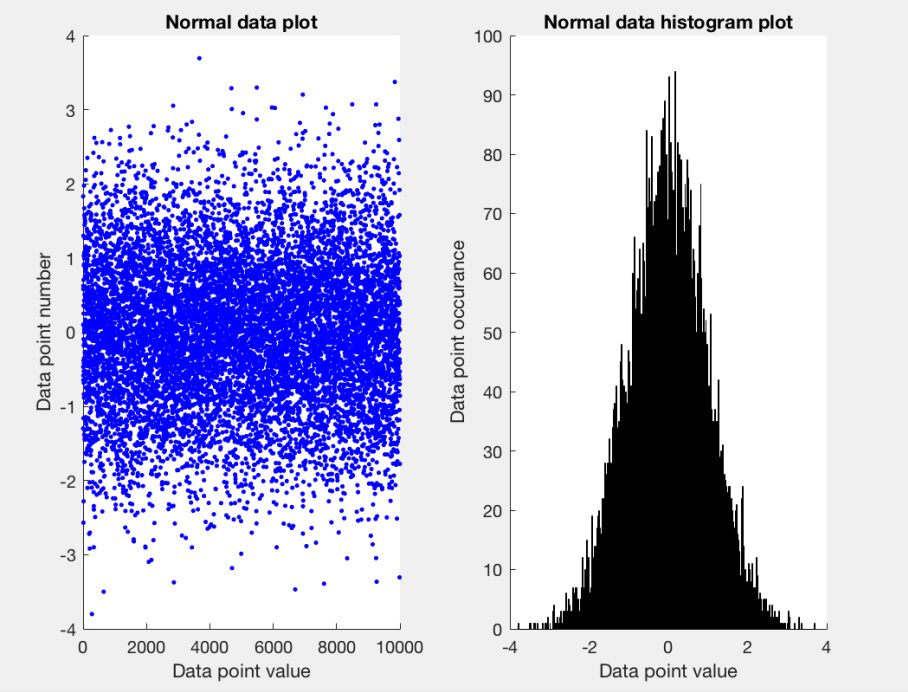
```
%Number of datapoints
samples = 10000;

%Create a Matrix of samples (nxn) from a uniform distribution
data = rand(samples);

%Get the mean of the 2nd dimension of the data set
meanData = mean(data,2);

%Specify the number of bins
nbins = 400;

%Print the matrix size
message = sprintf('data Matrix size %d x %d', size(data,1), size(data,2));
fprintf(message);

%Plot data on figure position 1
figure
subplot(1,2,1);
hold on
h = plot(meanData,'b.');
set(h,'linewidth',3);
title('Data mean plot');
xlabel('Data point number');
ylabel('Data point value');

%Plot histogram data on figure position 2
subplot(1,2,2);
hold on
histogram(meanData,nbins);
title('Data mean plot on histogram');
xlabel('Data point number');
ylabel('Data point occurance');
```



Student Number: 10450559

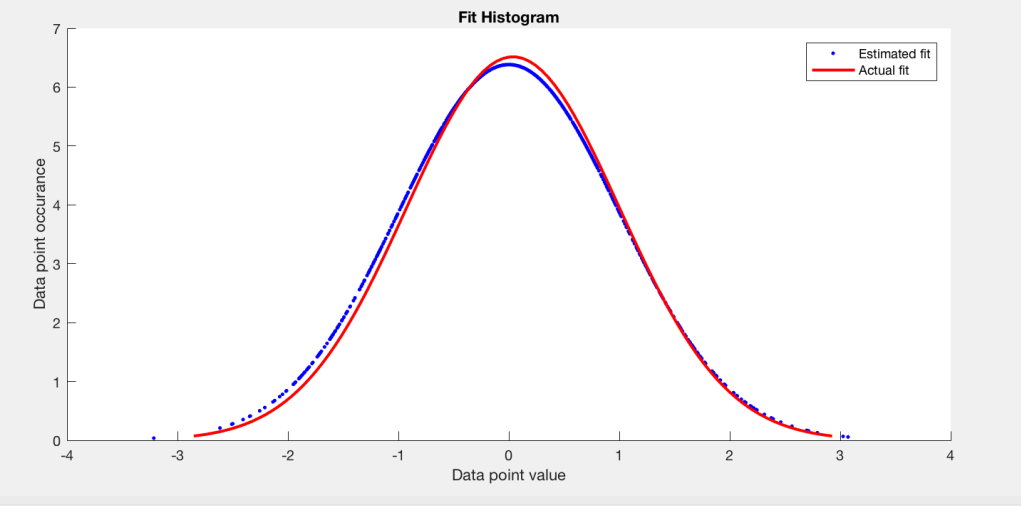
3 These results show a normal probability distribution. In this case a 1xn matrix is created from a sample of data from Matlab. By using the randn function this specifies to take the data from Matlabs in build set of normally distributed data set. This plot is similar to that of the previous central limit theorem but the mean does not need to be calculated manually, the data from the matrix generated by the randn function is simply plotted onto both the scatter plot and histogram. This shows that as the data approaches the mean value it starts to grow and then peak as it falls back down as it strays from the mean value. The larger the samples size used the clearer the normal distribution becomes. Similarly, with the increase of samples the increase of bins for the histogram will also make the normal distribution more clear.

```
 %Number of datapoints
samples = 10000;

%Create a Matrix of 1 X samples (1xn) from a normal distributon
data = randn(1,samples);

%Specify the number of bins
nbins = 400;

%Print the matrix size
message = sprintf('data Matrix size %d x %d', size(data,1), size(data,2));
fprintf(message);

%Plot data on figure position 1
figure
subplot(1,2,1);
hold on
h = plot(data,'b.');
set(h,'linewidth',3);
title('Normal data plot');
xlabel('Data point value');
ylabel('Data point number');

%Plot data on figure position 2
subplot(1,2,2);
hold on
i = histogram(data,nbins);
title('Normal data histogram plot');
xlabel('Data point value');
ylabel('Data point occurance');
```



4 The same as before this graphs shows the normal distribution along with the estimated values for the distribution. The actual distribution line has been fitted to the histogram where an increase in samples and bins will result in a more well fitted line and a better representation of the normal distribution. This plot is also known as a Gaussian distribution. By estimating the values of the mean and variance (square root of the standard deviation) we can estimate the plot and fitted line that the normal distribution will create. In order to have the estimation fit as closely as possible to the actual fit we have to scale the data appropriately. Otherwise the estimated plotted data will not provide much value when compared against the actual fit. Here the normal probability distribution is calculated (with the estimated variance and mean) and this is plotted against the data generated and once scaled will give an accurate representation of the normal distribution/ Gaussian plot.

```
%Number of datapoints
samples = 1000;

%Create a Matrix of 1 X samples (1xn)
data = randn(1,samples);

%Specify the number of bins
nbins = 400;

%the normal probability density function set mean to 0 and standard deviation to 1
norm = normpdf(data,0,1)*16;

%Split the data using the automatic algorithm
N = histcounts(data);

%Print the matrix size
message = sprintf('data Matrix size %d x %d', size(data,1), size(data,2));
fprintf(message);

%Plot the norm and the histogram with a fitted line on a figure.
figure
hold on
plot(data,norm,'b.');
h = histfit(data,nbins);
title('Fit Histogram');
ylabel('Data point occurance');
xlabel('Data point value');
```



5 In order to create a 2D distribution plot a 2D matrix is needed. Therefore, we again use the randn function but this time to create a 2xn matrix with a normal set of data. The randn function has a default standard deviation of 1 and mean of 0 which is applied to the 2D matrix. In order to represent this data, we must plot the dimensions against each other. To do this we take all of the first dimension as either x or y and all of the second dimension as either x or y. From the plot (once x and y are plotted against each other) we can see that the distribution is very focused on 0 on both access this is due to the mean of 0 and the set of normal data which is applied by the randn function when the matrix was created. An increase in samples will result on a more clustered distribution allowing the focus of 0 to become more clear.

```
1     %Number of datapoints
2 -   samples = 10000;
3
4     %Create a Matrix of 2 X samples (2xn) (randn is produced with a mean of 0
5     %and standard deviation of 1, otherwise it would be: standardDeviation * randn(2,n) + mean;)
6 -   data = randn(2,samples);
7
8     %Set the x and y points to be plotted. One dimension against another.
9 -   x = data(1,:);
10 -  y = data(2,:);
11
12    %Print the matrix size
13 -  message = sprintf('data Matrix size %d x %d', size(data,1), size(data,2));
14 -  fprintf(message);
15
16    %Plot data on figure position 1
17 -  figure
18 -  subplot(1,1,1);
19 -  hold on
20 -  scatter(x,y,'b.');
21 -  title('Uniform 2D plot');
22 -  xlabel('x-value');
23 -  ylabel('y-value');
```