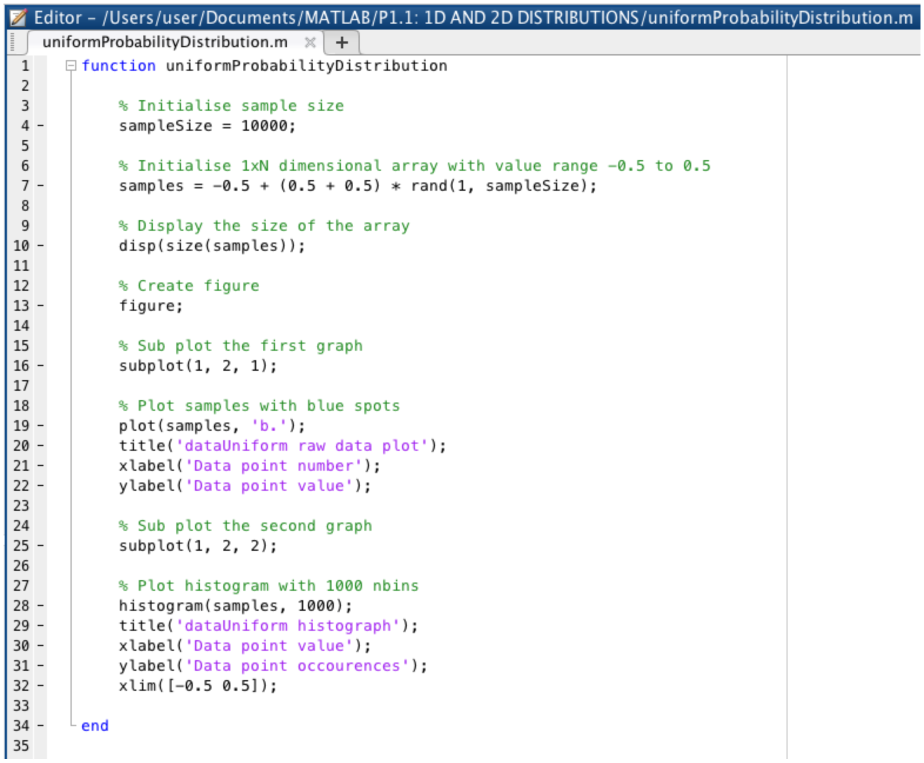
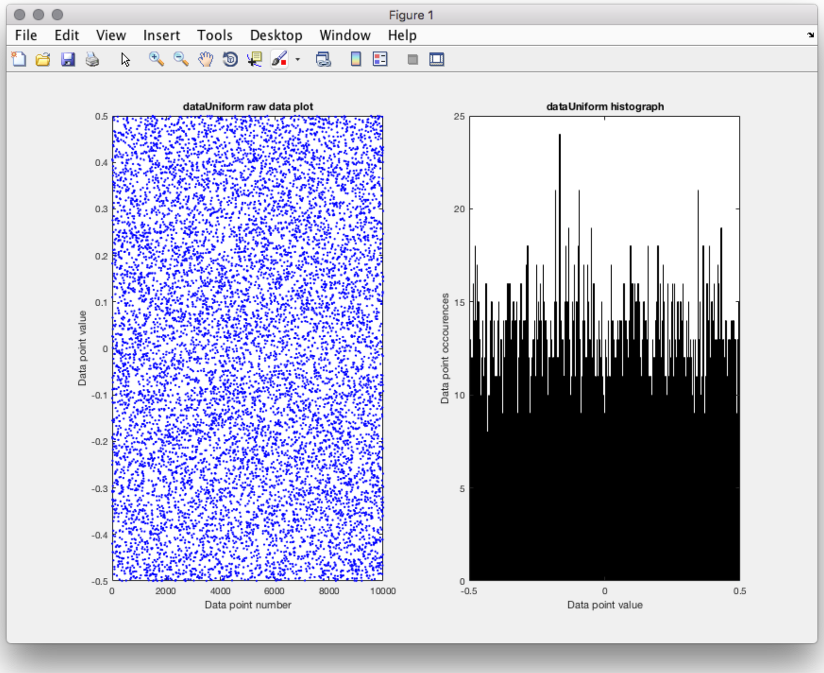
AINT351 - P1.1 1D and 2D distributions

1. Generate a uniform probability distribution


Editor – /Users/user/Documents/MATLAB/P1.1: 1D AND 2D DISTRIBUTIONS/uniformProbabilityDistribution.m
uniformProbabilityDistribution.m
+

```matlab
function uniformProbabilityDistribution

    % Initialise sample size
    sampleSize = 10000;

    % Initialise 1xN dimensional array with value range -0.5 to 0.5
    samples = -0.5 + (0.5 + 0.5) * rand(1, sampleSize);

    % Display the size of the array
    disp(size(samples));

    % Create figure
    figure;

    % Sub plot the first graph
    subplot(1, 2, 1);

    % Plot samples with blue spots
    plot(samples, 'b.');
    title('dataUniform raw data plot');
    xlabel('Data point number');
    ylabel('Data point value');

    % Sub plot the second graph
    subplot(1, 2, 2);

    % Plot histogram with 1000 nbins
    histogram(samples, 1000);
    title('dataUniform histograph');
    xlabel('Data point value');
    ylabel('Data point occourences');
    xlim([-0.5 0.5]);

end

```

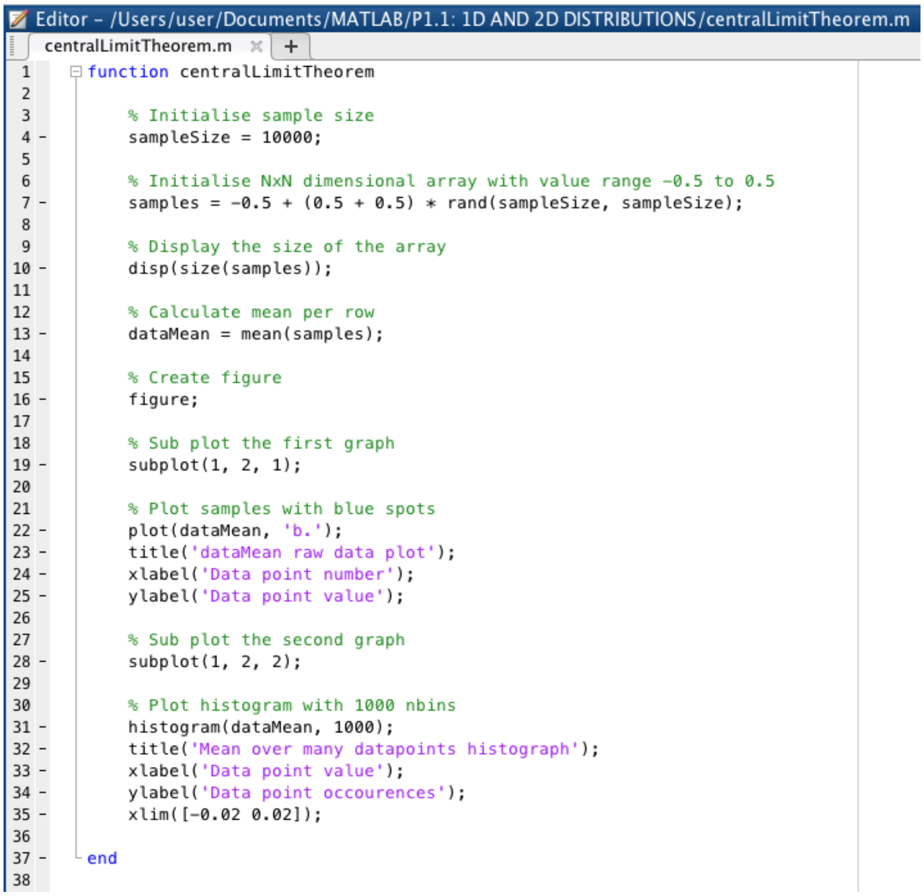


The task was to create a uniform probability distribution graph which demonstrates the theory that all data points that are equal in number of occurrences, have an equal probability to be chosen. This is partly because we fill the matrix with data, no other calculations, therefore all numbers is likely to appear the same amount of times.

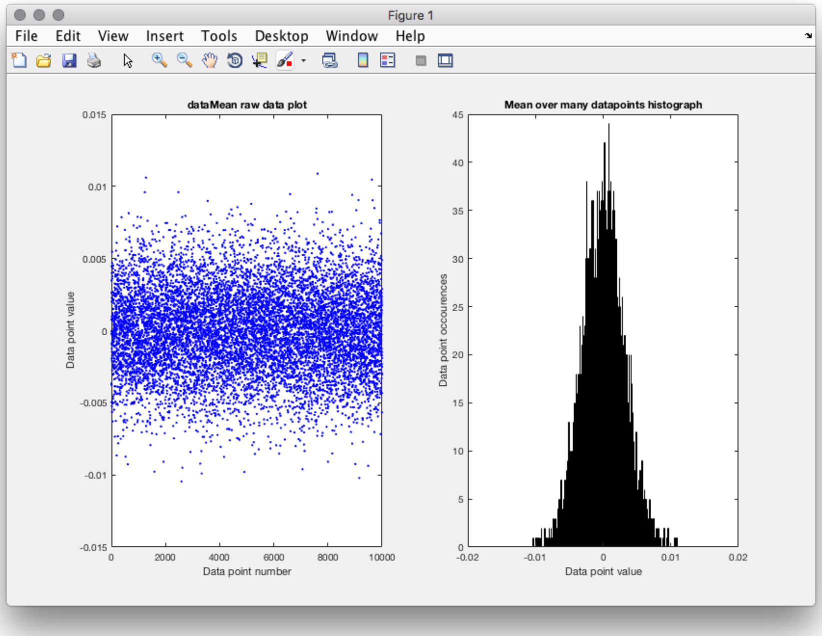
Changing the number of bins increases the size of the intervals on the x-axis, meaning that the data point values have a larger group to fall in. This pushes the occurrence of each bin much higher the less bins there are. If I were to change the bin from 1000 to 100 then the data point occurrence range goes from the maximum of 25 to 140. This, effectively, decreasing the accuracy. It also means that the peaks and dips of the ranges have a greater difference between them.

The number of samples that have been used has been consistent throughout all graph generations, standing at 10,000 values. This is large enough for there it to have a good range of values even at lower levels of bins but not high enough so that the generation of the graph takes too long nor that the graph is over saturated with values that are unnecessary to prove the point of the graph.

2. The central limit theorem


Editor - /Users/user/Documents/MATLAB/P1.1: 1D AND 2D DISTRIBUTIONS/centralLimitTheorem.m
centralLimitTheorem.m
+

```matlab
function centralLimitTheorem

    % Initialise sample size
    sampleSize = 10000;

    % Initialise NxN dimensional array with value range -0.5 to 0.5
    samples = -0.5 + (0.5 + 0.5) * rand(sampleSize, sampleSize);

    % Display the size of the array
    disp(size(samples));

    % Calculate mean per row
    dataMean = mean(samples);

    % Create figure
    figure;

    % Sub plot the first graph
    subplot(1, 2, 1);

    % Plot samples with blue spots
    plot(dataMean, 'b.');
    title('dataMean raw data plot');
    xlabel('Data point number');
    ylabel('Data point value');

    % Sub plot the second graph
    subplot(1, 2, 2);

    % Plot histogram with 1000 nbins
    histogram(dataMean, 1000);
    title('Mean over many datapoints histograph');
    xlabel('Data point value');
    ylabel('Data point occourences');
    xlim([-0.02 0.02]);

end

```

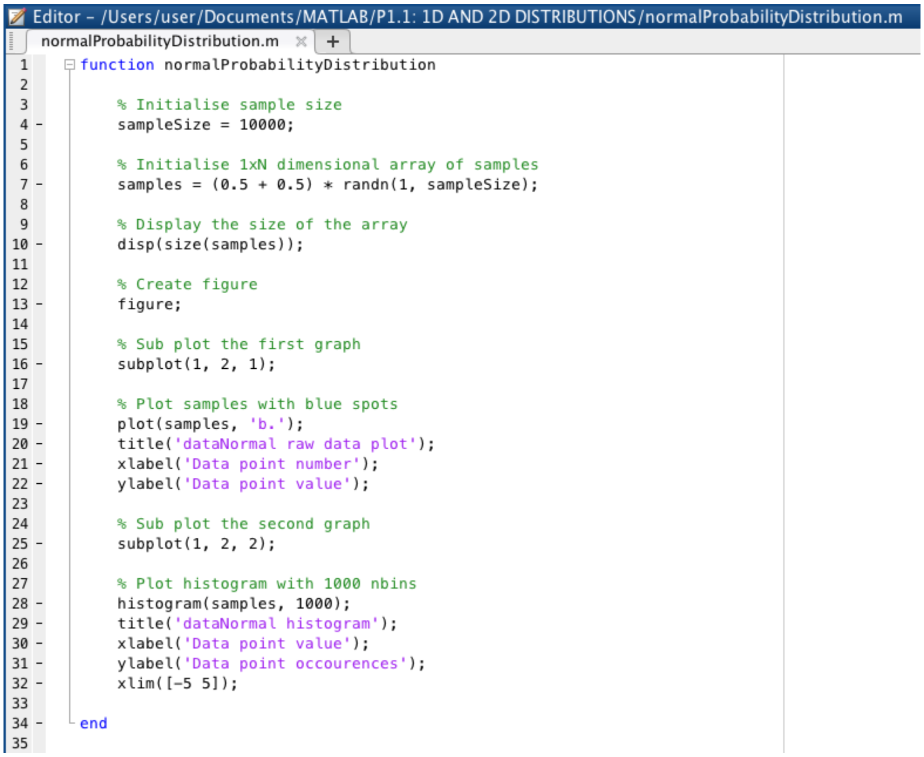


The task was to create a central limit theorem graph which demonstrates the theorem that the mean of a large number of iterations of data that are independent, will approximately be normally distributed. That is, no matter data distribution, the mean will gravitate toward the middle point of the data and will have a greater probability than that of the data closer to the limits of data boundaries.

Increasing the size of data used increases the height of the centre of the distribution, where as decreasing not only shortens the height of the centre but widens the standard deviation from the centre of the distribution. Keeping the same number of samples but decreasing the number of bins from 1000 to 10 dramatically changes the appearance of the graph. Although the rise of the curve from outer limits to the centre limit is already steep, decreasing the number of bins only emphasises that feature. As always, it pushes the data occurrences limit up greatly due to there being more chance of falling in a bin.

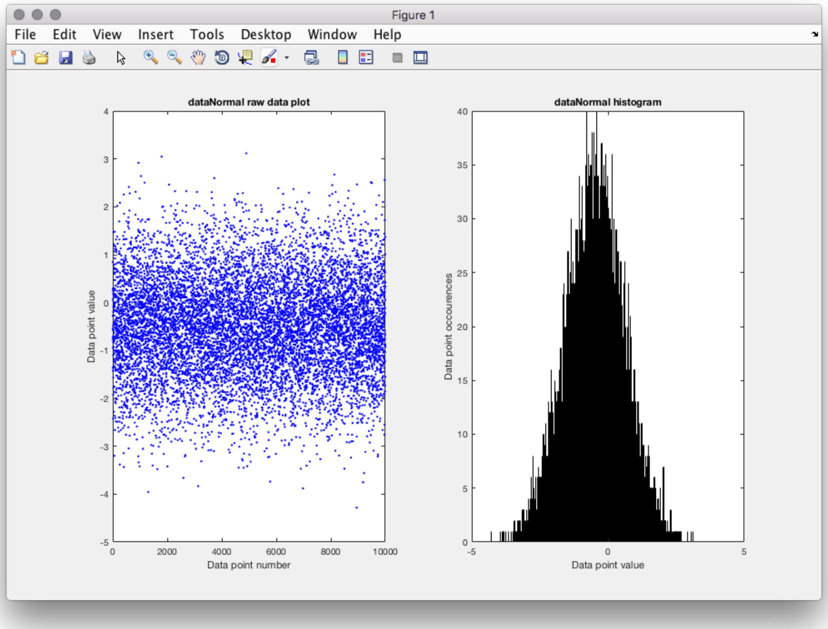
I think that 10,000 adequately show the purpose of the graph to a good degree of accuracy.

3. Generate a normal probability distribution



Editor – /Users/user/Documents/MATLAB/P1.1: 1D AND 2D DISTRIBUTIONS/normalProbabilityDistribution.m
normalProbabilityDistribution.m
+

```
function normalProbabilityDistribution

    % Initialise sample size
    sampleSize = 10000;

    % Initialise 1xN dimensional array of samples
    samples = (0.5 + 0.5) * randn(1, sampleSize);

    % Display the size of the array
    disp(size(samples));

    % Create figure
    figure;

    % Sub plot the first graph
    subplot(1, 2, 1);

    % Plot samples with blue spots
    plot(samples, 'b.');
    title('dataNormal raw data plot');
    xlabel('Data point number');
    ylabel('Data point value');

    % Sub plot the second graph
    subplot(1, 2, 2);

    % Plot histogram with 1000 nbins
    histogram(samples, 1000);
    title('dataNormal histogram');
    xlabel('Data point value');
    ylabel('Data point occourences');
    xlim([-5 5]);

end

```

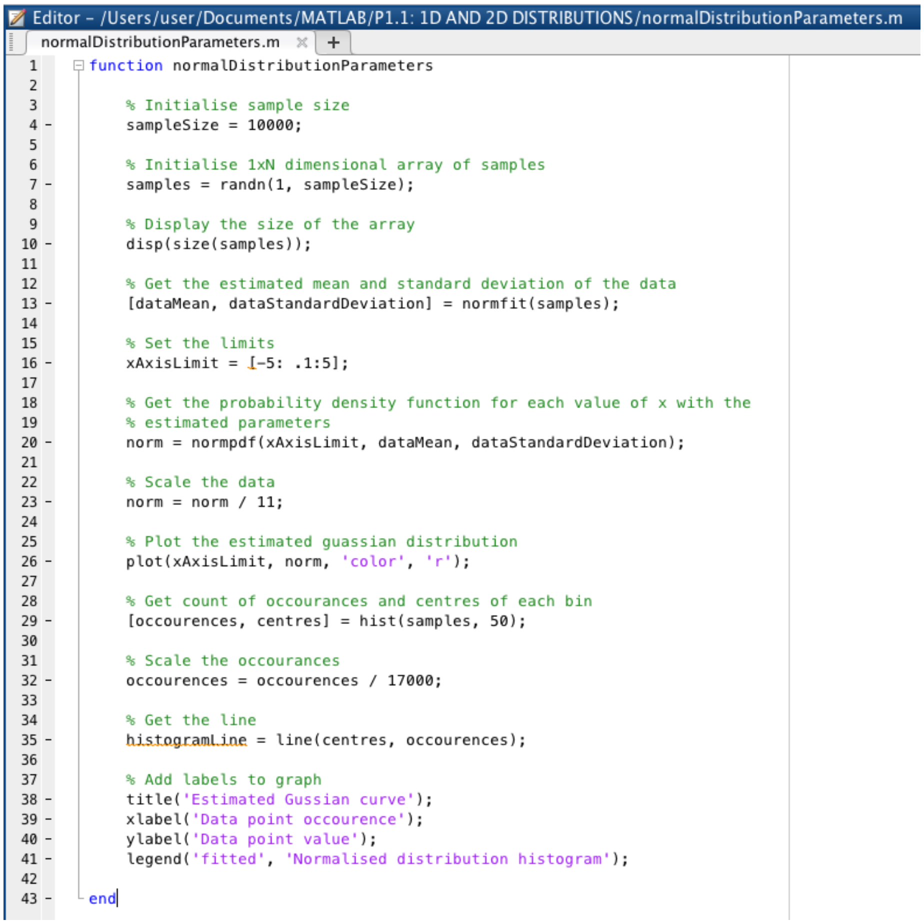


The task was to create a normal probability distribution graph which demonstrates that the data mean is greatest at its median. Many categories of data fall into this distribution such as average height, average weight, etc. This graph was generated like this because of the use of randn function, which generates randomly normally distributed numbers.

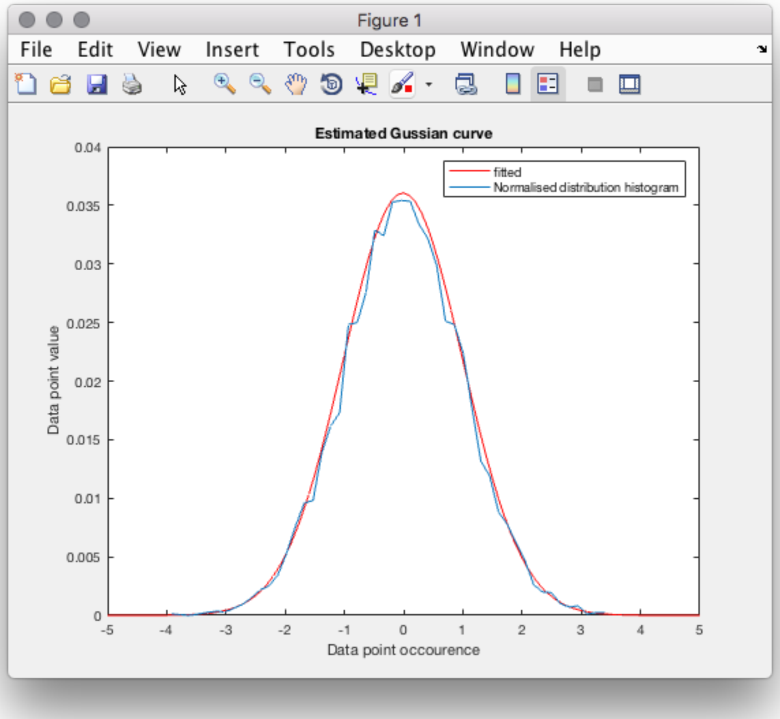
If I were to double the amount of samples used from 10,000 to 20,000, the occurrence of the data at the median greatly increases, almost doubles in fact, but the graph does not widen. The same is for the decreasing of the number of bins, it does not widen the graph, but only heighten the peak at the median. This is because I've restricted the range in which the random numbers were generated in to 5 and -5, so it only further proves that adding more data to such distribution only strengthens the concept that it the mean of it all is falls close to the median.

I believe that between 10,000 and 20,000 is more than enough data points to show the distribution's main characteristics.

4. Estimate a normal distribution parameters


Editor – /Users/user/Documents/MATLAB/P1.1: 1D AND 2D DISTRIBUTIONS/normalDistributionParameters.m
normalDistributionParameters.m
+

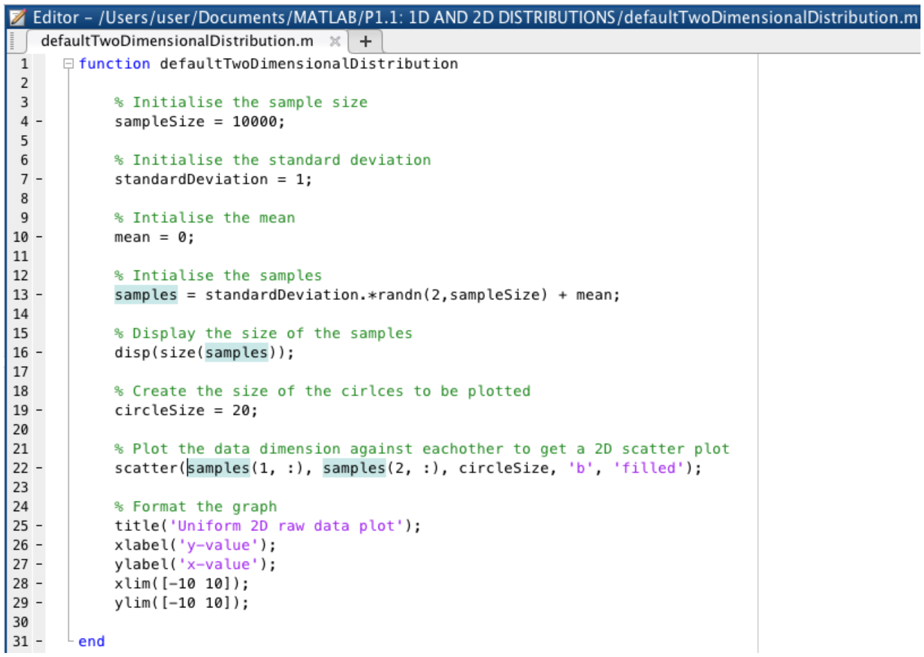
```
function normalDistributionParameters

    % Initialise sample size
    sampleSize = 10000;

    % Initialise 1xN dimensional array of samples
    samples = randn(1, sampleSize);

    % Display the size of the array
    disp(size(samples));

    % Get the estimated mean and standard deviation of the data
    [dataMean, dataStandardDeviation] = normfit(samples);

    % Set the limits
    xAxisLimit = [-5: .1:5];

    % Get the probability density function for each value of x with the
    % estimated parameters
    norm = normpdf(xAxisLimit, dataMean, dataStandardDeviation);

    % Scale the data
    norm = norm / 11;

    % Plot the estimated guassian distribution
    plot(xAxisLimit, norm, 'color', 'r');

    % Get count of occourances and centres of each bin
    [occourences, centres] = hist(samples, 50);

    % Scale the occourances
    occourences = occourences / 17000;

    % Get the line
    histogramLine = line(centres, occourences);

    % Add labels to graph
    title('Estimated Gussian curve');
    xlabel('Data point occourence');
    ylabel('Data point value');
    legend('fitted', 'Normalised distribution histogram');

end
```



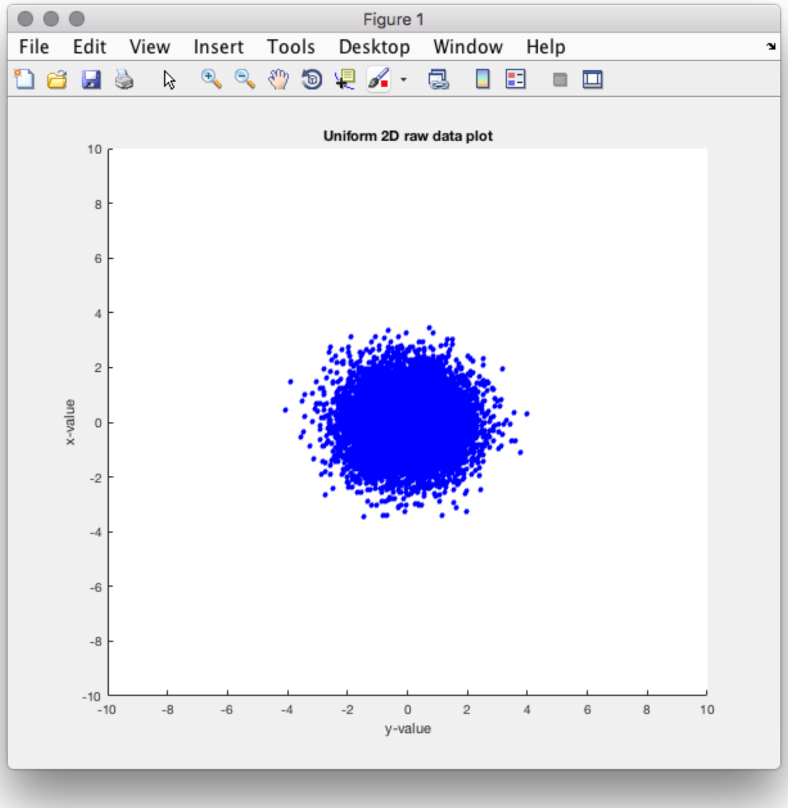
The task was to plot a scaled version of estimated mean and variance of the samples against a scaled normal distribution probability graph. As explained previously, the normal probability distribution graph shows that when data is normally distributed, its mean is greatest at the median, giving the Guassian or "bell" shaped curve when plotted. The estimated Guassian distribution was created using MatLab's "normfit()" function which returns the estimated mean and standard variance of the data passed in. Then the data had to be scaled to fit in the data occurrence scale provided. I achieved the right scaling parameters purely by trail and error, I was unsure if there was a formula that gives the right scaling values for both graphs for them to match up. The estimated Guassian line gives a line of best fit on the raw data samples when they are scaled the same.

Decreasing the total number samples used doesn't effect the estimated Guassian graph greatly, but it dramatically decreases the height of the data point occurrences of the raw data. This is because the calculation to find the estimated mean remains the same but the number of data that could possibly occur reduces giving a lower total spread across all data values. Having the sample size set to 10,000 means the graphs curve is of a size that demonstrates the point with clarity.

5. Generate a default 2D distribution


Editor - /Users/user/Documents/MATLAB/P1.1: 1D AND 2D DISTRIBUTIONS/defaultTwoDimensionalDistribution.m
defaultTwoDimensionalDistribution.m
+

```
function defaultTwoDimensionalDistribution

    % Initialise the sample size
    sampleSize = 10000;

    % Initialise the standard deviation
    standardDeviation = 1;

    % Intialise the mean
    mean = 0;

    % Intialise the samples
    samples = standardDeviation.*randn(2,sampleSize) + mean;

    % Display the size of the samples
    disp(size(samples));

    % Create the size of the cirlces to be plotted
    circleSize = 20;

    % Plot the data dimension against eachother to get a 2D scatter plot
    scatter(samples(1, :), samples(2, :), circleSize, 'b', 'filled');

    % Format the graph
    title('Uniform 2D raw data plot');
    xlabel('y-value');
    ylabel('x-value');
    xlim([-10 10]);
    ylim([-10 10]);

end
```



The task was to graph a representation of a default 2D distribution, which, in essence, is a normal probability distribution but of a higher dimension. At its heart, it derives from the central limit theorem as it shows two independently correlated set of random normal distributed data's mean still relatively equal to the median of the data values.

If you were to reduce the number of samples, it reduces the density of the cluster in the middle and makes the results more sparse. Increasing the standard deviation increases the distance of the data's relation to the mean. Changing the value of the mean shifts the middle of the cluster to the value set.

TO REMEMBER

- understanding of either rand or randn
- understanding of mean var

- insight into task
- (matlab code)
- What is the task
- How i solve it
- What does it mean
- **Eplain**
- How does changing number samples, bins effect what you see
- Guassian adding no tegether, doesn't matter what limit they tend to go to a guassain form
- what constitutes a sensible choice
- model data by estimate parameters

1. Generate a noisy line