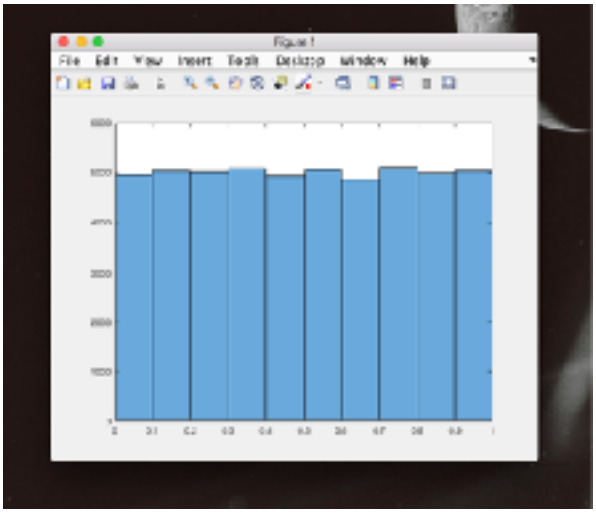
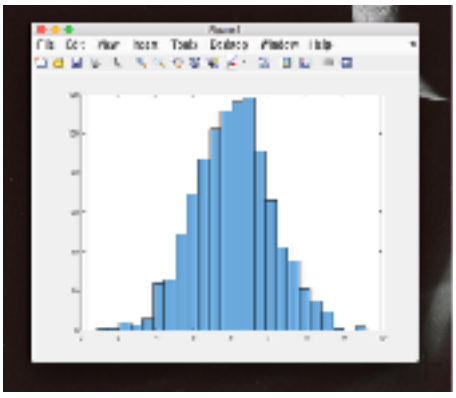
Corbin Muraro
Homework 1

1)

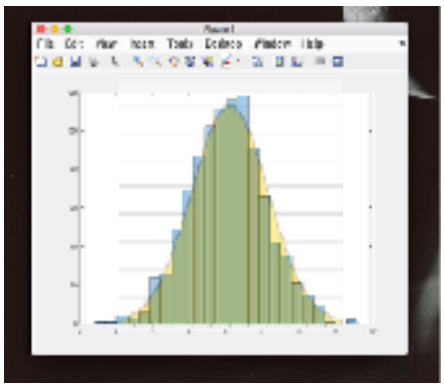
a) I used the rand() function.
b) I generated 50,000 random numbers via rand()
c) I plotted the 50,000 numbers with the histogram() function. Here is the result:



2) I generated the following histogram:



I then overlaid a true normal curve I found online over the randomly generated random curve compare the area covered by my randomly generated normal curve. The middle 20% of the generated graph was responsible for 430 / 1000 of the elements, or 43%. A true normal curve contains 38% of the elements in the middle 20% of the curve.



3)

a) Because the vectors are normalized, the dot product represents the angle between the vectors, or the degree of similarity between the directions of the vectors

b) Output:

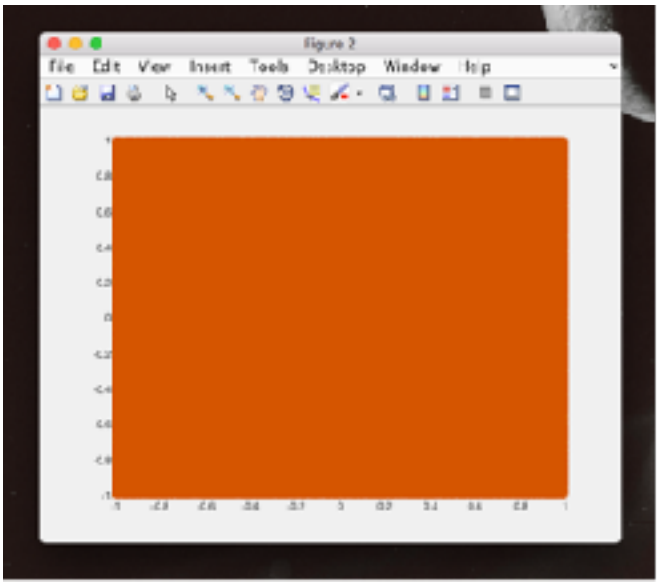
```
>> hw1_3
Dimension: 10, Mean: 0.008153, Standard Deviation: 0.316938
Dimension: 20, Mean: -0.002674, Standard Deviation: 0.224426
Dimension: 50, Mean: -0.001628, Standard Deviation: 0.143723
Dimension: 100, Mean: -0.001585, Standard Deviation: 0.099969
Dimension: 250, Mean: 0.001038, Standard Deviation: 0.062822
Dimension: 500, Mean: -0.000297, Standard Deviation: 0.044175
Dimension: 1000, Mean: 0.000081, Standard Deviation: 0.031658
Dimension: 2000, Mean: 0.000245, Standard Deviation: 0.022541
```

c) The mean should approach zero, as the vectors are approaching linear independence as the dimensionally increases. Standard deviation should also approach zero as sample size increases. Nonzero values are purely because of the variance inherent to random number generation.

d) Using the formula for expected value and variance, I found standard deviation to be the 1 / sqrt(# dimensions). This agrees with the results I got, and after looking at Matlab results, the same appears to hold true for higher dimensions.

4) With 10,000,000 random points plotted in the graph from y = –1 to y = 1 and x = –1 to x = 1, 7,854,263 were found to land within the unit circle.

Dividing 4 by the actual value of Pi = 1.27324
Meanwhile, my estimate, by dividing all points inside the unit circle by all total points in the graph from y = –1 to y = 1 and x = –1 to x = 1, or 10,000,000 / 7,854,263 = 1.27319. With 10,000,000 random points, the accuracy of the approximation becomes very high.



(for some reason, the unit circle isn't showing through the color of the plot anymore, but the circle is still plotted)