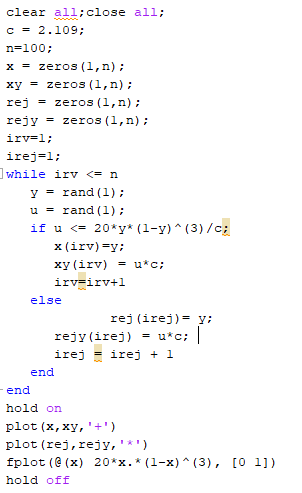
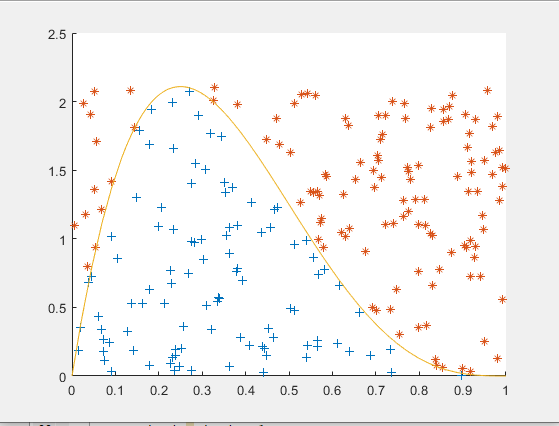
# In Class

Problem Statement: Use the accept/reject method to generate a sample from a beta distribution. Include plot of samples showing candidates accepted and those rejected.

Code



Plot



Discussion

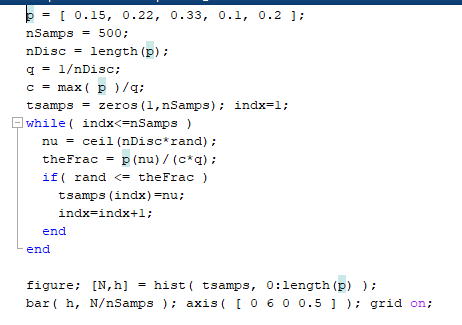
The domain of the beta distribution is (0,1) so the distribution we use to generate our first sample is the Uniform(0,1). We find a constant C to inflate the uniform distribution. This c was found to be 2.109, the maximum value that the beta distribution we are working with can take. With this value we set up the accept/reject algorithm and produce a plot showing all values that were accepted and all that were rejected. We have also plotted the beta function to visualize the algorithm’s success.

# 4.2

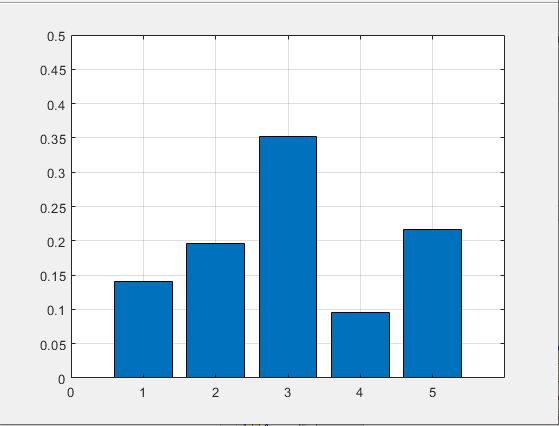
Problem Statement

Write Matlab code to implement example 4.5. Generate 400 random variables and construct a histogram to verify.

Code



Plot



Discussion

The empirical frequencies match very closely with the probabilities of the distribution given

P(X= 1) = 0.15

P(X= 2) = 0.22

P(X= 3) = 0.33

P(X= 4) = 0.10

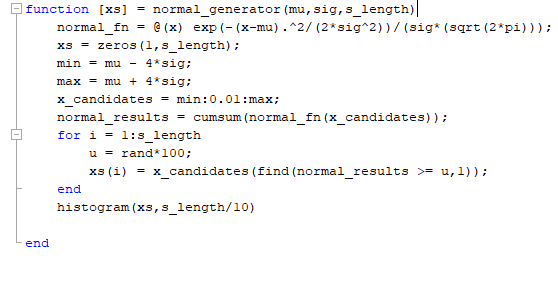
P(X= 5) = 0.20

# 4.5

Problem Statement:

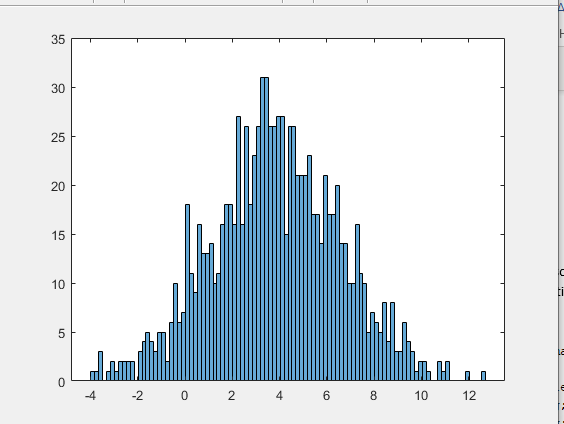
Use the alternative method discussed in class to generate random numbers from the normal distribution. Write it as a function that takes the arguments for mu and sigma.

Code



Plot





Discussion

The function works by creating a very long vector of possible values the random variable could take. The next step is to find a probability for each of those possible values, and use them to create an array of CDF values that is 1:1 indexed with the correct values the variable can take. A loop then creates a random number (Uniform(0,1)\*100) and compares that the values in the CDF array. The index of the smallest value that is greater than the random number is returned and used to select the corresponding value from the possible values.

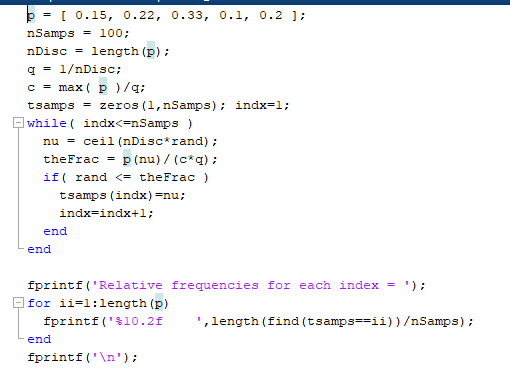
The histrogram created shows 1000 samples from a Normal(4,3) distribution.

# 4.9

Problem Statement

Implement example 4.5 again to generate 1000 random numbers. Check that their relative frequency matches with the probability function

Code



Discussion



The empirical frequencies match very closely with the probabilities of the distribution given

P(X= 1) = 0.15

P(X= 2) = 0.22

P(X= 3) = 0.33

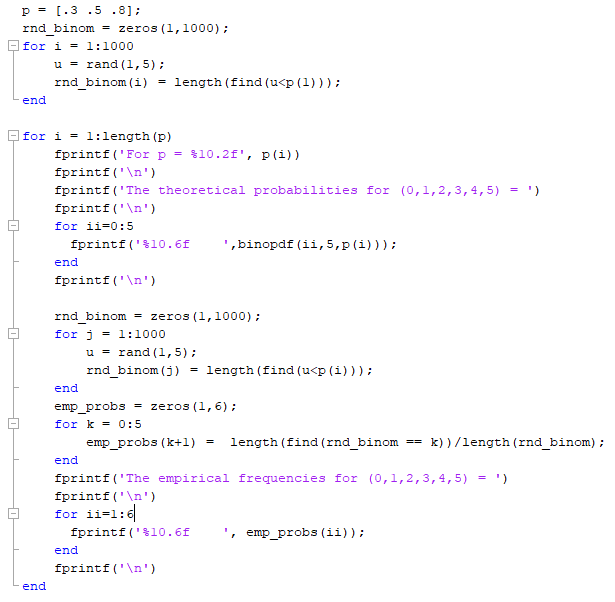
P(X= 4) = 0.10

P(X= 5) = 0.20

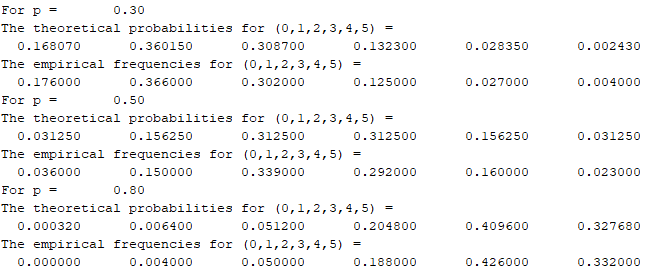
# 4.12

Problem Statement: Generate 1000 binomial random variables for n = 5 and p = 0.3,0.5,0.8

Code



Results



Discussion

With the sample size of 1000, the empirical frequencies match very closely with the theoretical probabilities of the Binomial distribution.