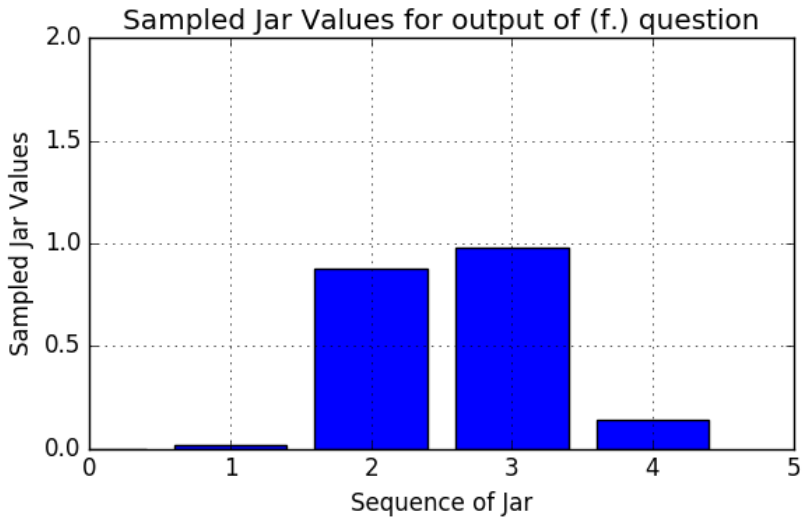
Homework 3

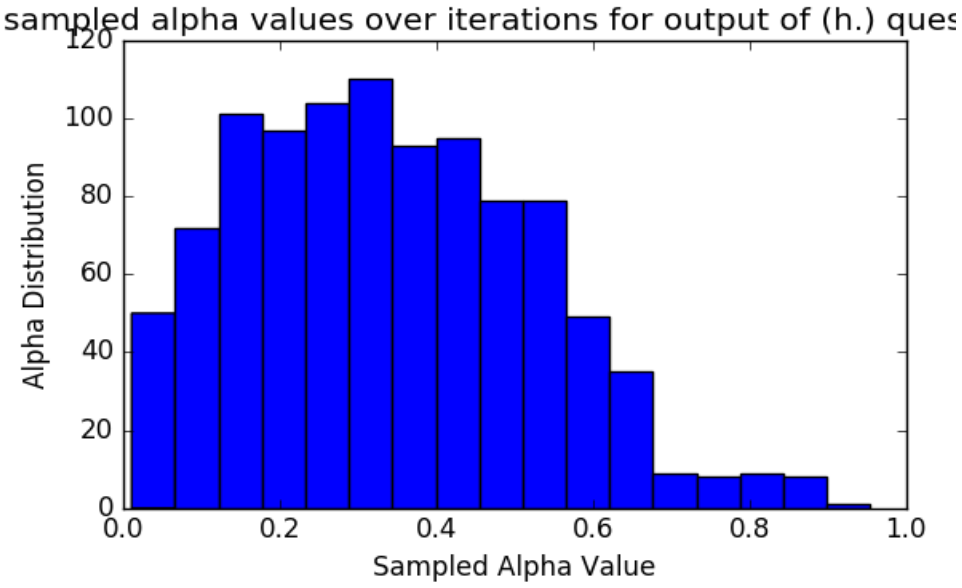
1.

(f). $\boldsymbol{J_{sampled}}$ = [0.0, 0.22, 0.76, 0.92, 0.18]



Figure(f.) Indicating Sampled Jar Values using Metropolis Algorithm

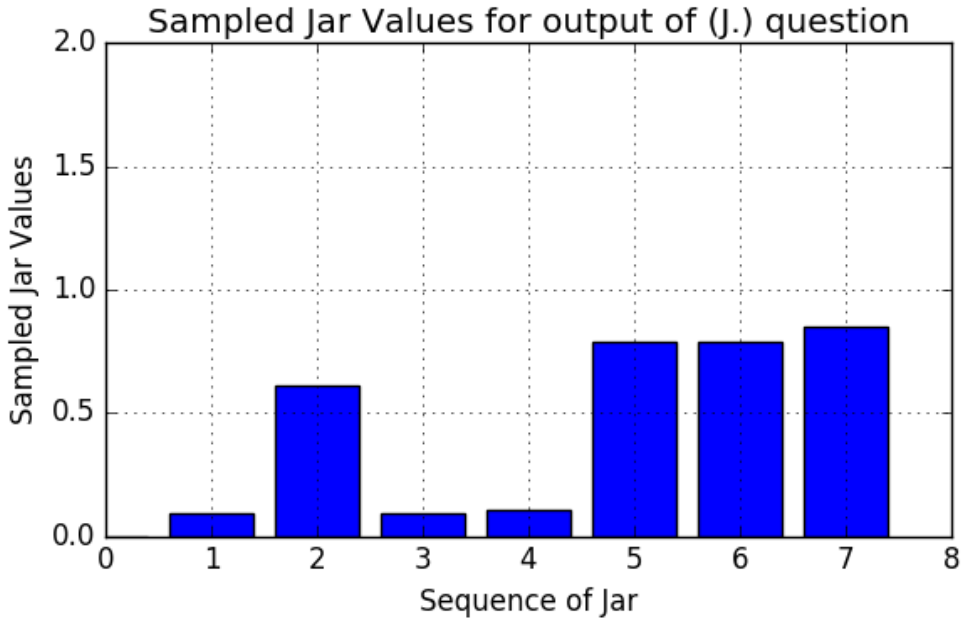
(h.) $\boldsymbol{\alpha_{mean}}$= 0.33



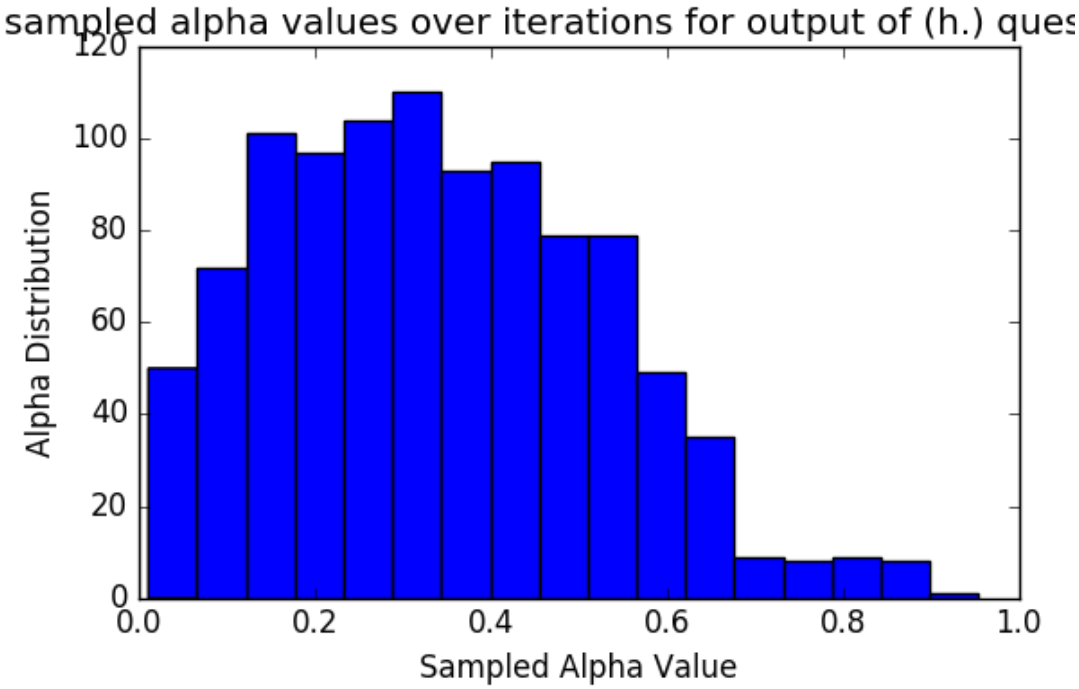
Figure(h.) Indicating Sampled Alpha values with Y axis representing the number of occurrences of certain alpha value, and X axis representing that value.

(j.) $\alpha_{mean} = 0.597$

$J_{sampled}$ =[0.0, 0.091, 0.614, 0.0919, 0.106 0.791, 0.791, 0.851]



Fig(j(a)): Indicating Sampled Jar Values using Metropolis Algorithm



Fig(j(b)): Indicating Sampled Alpha values with Y axis representing the number of occurrences of certain alpha value, and X axis representing that value.

2. Experimentation for Alpha Values:

Alpha mean value: 0.45

Proposal: Instead of using Uniform Distribution, use other Probability Distribution functions such as Normal, Laplace and Logistic Distribution.

Algorithm: Function Random Alpha(Alpha):

> **Get random alpha value of different probability distribution functions by setting mean as alpha.**

Observations: If we run Metropolis algorithm for alpha sampling multiple number of times, then it becomes independent of mean and standard deviation.

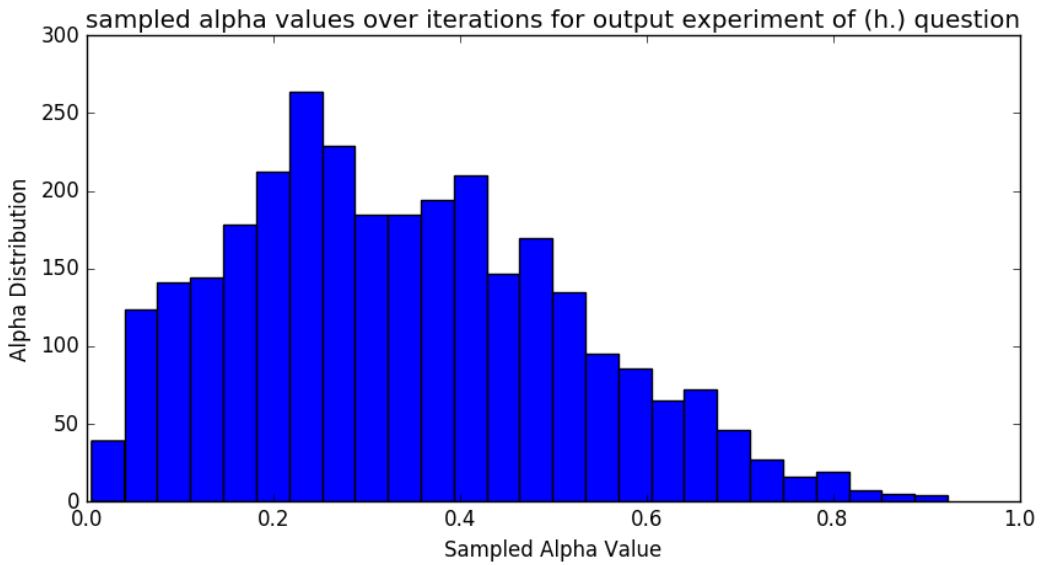


Figure: Alpha mean distribution with Normal Probability Distribution Function

Experimentation for Jar Values:

Proposal: Instead of flipping one bit, flip random number of bits, at max half of length of given Jar Sequence.

Algorithm (Pseudo Code):

Randomized J Values:

Limit= random integer value ranging from 1 to (length of J)/2

For values in (0, Limit):

> Choose a random position of J
>
> Flip the bit

Return the randomized J values.

Result: On implementation of this sampling, convergence of values was possible within 500 iterations with greater confidence most of times.

While when we were flipping one bit at a time, we were just considering one variable, but if we randomize the chosen number of flipped bits, we are considering all possible dimensions of sampling, and after that Metropolis Algorithm Markov chain helps us converging to a possible sampled J set, in less number of iterations.

Proof: output of 1.(f) question in 500 iterations.

Uniform 1 bit flipped J Sampled Output: [0.0, 0.22, 0.56, 0.77, 0.19]

Uniform random number of bits flipped J sampled output: [0.0, 0.12, 0.85, 0.87, 0.0]

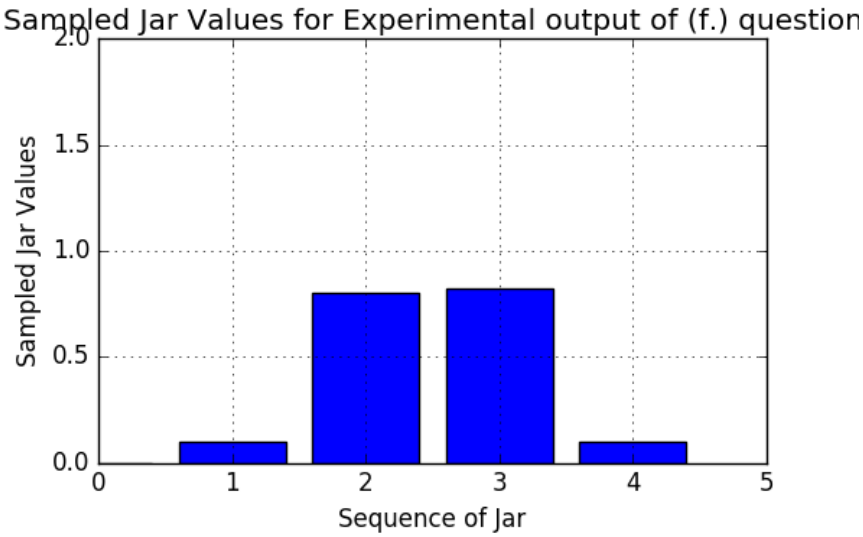


Figure: Running Randomized J value 500 times.