Midterm Exam

Problem 1

Given $X \sim U[-1, 1]$

$Y_1 = X = r(X)$; $Y_2 = 3X^2 - 1 = s(X)$

Part 1)

$$E[Y_1] = \int_{-\infty}^{\infty} y_1 \cdot f(y_1) \cdot dy_1$$

$$= \int_{-\infty}^{\infty} r(x) \cdot f(x) \cdot dx$$

From lecture notes, we know that

$$\int_{D_Y} y \cdot f_Y(y) \cdot dy = \int_{D_X} g(x) \cdot f_X(x) \cdot dx$$

$$= \int_{-1}^{1} x \cdot f(x) \cdot dx$$

$$= \int_{-1}^{1} x \cdot \left(\frac{1}{2}\right) \cdot dx$$

$$= \frac{1}{2} \int_{-1}^{1} x \cdot dx$$

$$= \frac{1}{2} \cdot \frac{1}{2} \cdot \left[x^2\right]_{-1}^{1}$$

$$= \frac{1}{4} \cdot \left(1^2 - (-1)^2\right)$$

$$= 0$$

$$E[Y_2] = \int_{-\infty}^{\infty} y_2 \cdot f(y_2) \cdot dy_2$$

$$= \int_{-\infty}^{\infty} s(x) \cdot f(x) \cdot dx$$

$$= \int_{-\infty}^{\infty} (3x^2 - 1) \cdot f(x) \cdot dx$$

$$= \int_{-1}^{1} (3x^2 - 1) \cdot \left(\frac{1}{2}\right) \cdot dx$$

$$= \frac{1}{2} \cdot \left[\int_{-}^{1} 3x^2 \cdot dx - \int_{-1}^{1} dx \right]$$

$$= \frac{1}{2} \left[x^3 \Big]_{-1}^{1} - x \Big]_{-1}^{1} \right]$$

$$= \frac{1}{2} \left[\left(1^3 - (-1)^3 \right) - \left(1 - (-1) \right) \right]$$

$$= \frac{1}{2} \left[2 - 2 \right]$$

$$= 0$$

Part 2) Covariance of Y_1 and Y_2 is defined as

$$C_{Y_1Y_2} = E\left[(Y_1 - E[Y_1]) (Y_2 - E[Y_2]) \right]$$

$\because$ $E[Y_1] = E[Y_2] = 0$, we get

$$C_{Y_1Y_2} = E\left[Y_1 Y_2 \right]$$

$$= E\left[X(3X^2-1) \right]$$

$$= \int_{-\infty}^{\infty} (3x^3 - x) \cdot f(x) \cdot dx$$

$$= \int_{-1}^{1} (3x^3 - x) \cdot \frac{1}{2} \cdot dx$$

$$= \frac{1}{2} \cdot \left[\int_{-1}^{1} 3x^3 \cdot dx - \int_{-1}^{1} x \cdot dx \right]$$

$$= \frac{1}{2} \cdot \left[\frac{3}{4} x^4 \Big]_{-1}^{1} - \frac{1}{2} x^2 \Big]_{-1}^{1} \right]$$

$$= \frac{1}{2} \cdot \left[\frac{3}{4} \cdot (1-1) - \frac{1}{2}(1-1) \right]$$

$$= 0$$

Since $C_{Y_1Y_2} = 0$, we can conclude that Y_1 and Y_2 are uncorrelated.

' N

Problem 2

Proposed Approach :

1. Find min and max integers between which the given data lies.

2. Generate bins that are 1 unit apart and count the number of points in each bin.

3. Divide the count of each bin ~~by~~ by 200 to get probability value for that bin.

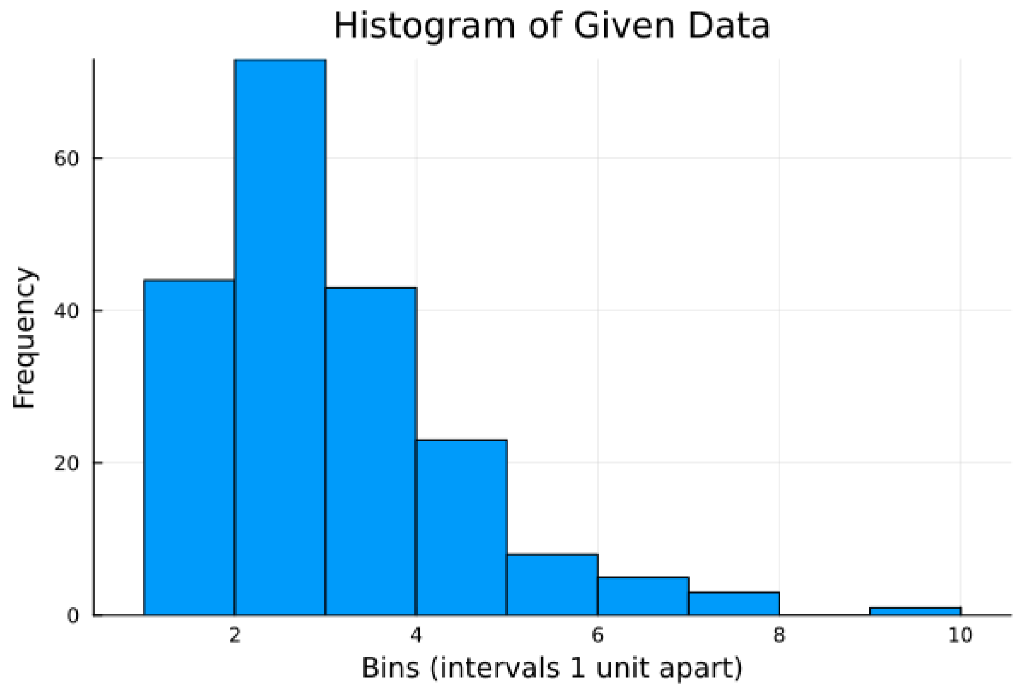
4. Now, to sample 1000 points

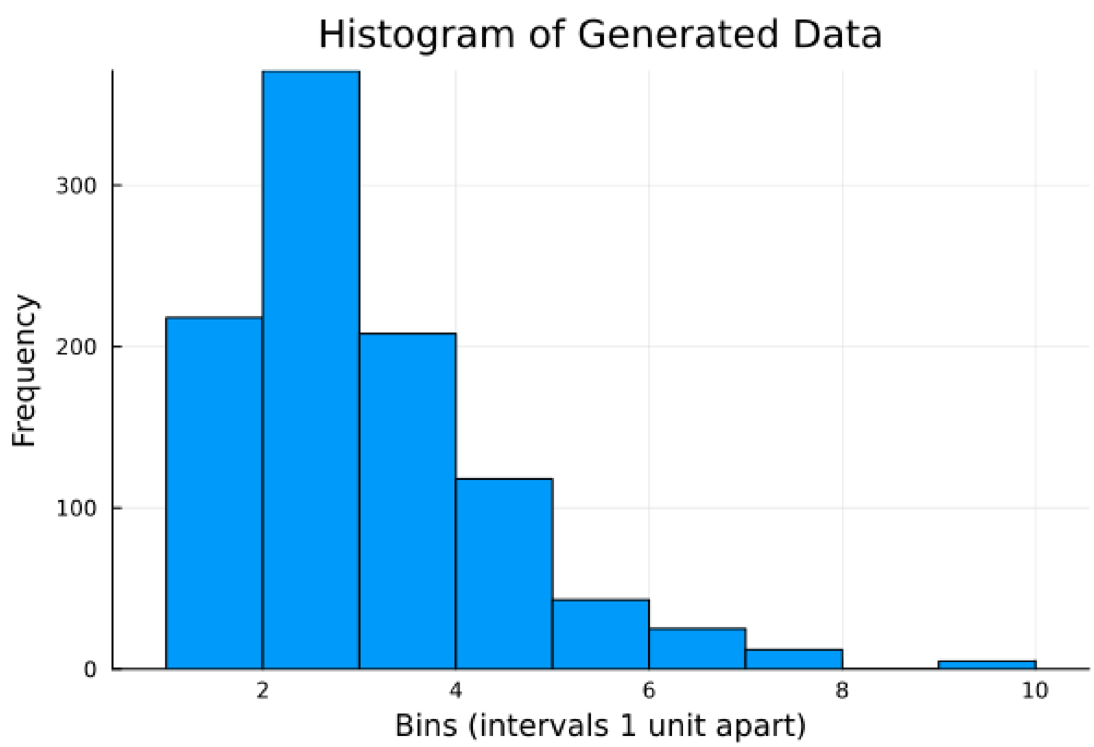
   a) Sample a bin using the discrete probability values obtained in step 3.

b) Sample a value uniformly from that bin.

Generated histograms for given and generated data are attached below.
They are quite similar to each other.

Plots for Problem 2





Problem 3

Total Possible outcomes $\longrightarrow$ $6 \times 6 = 36$

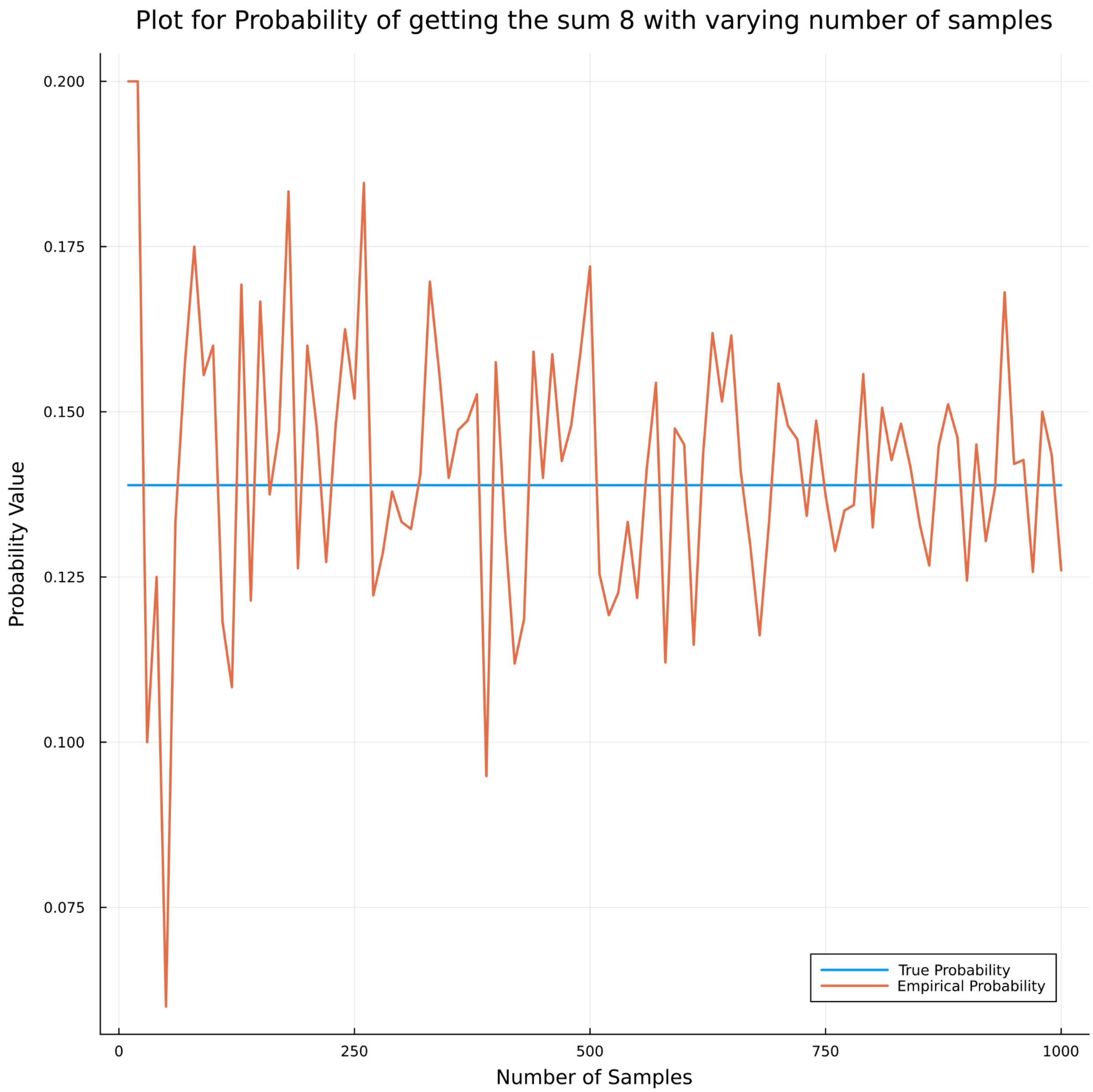
Total outcomes with sum 8 $\longrightarrow$ 5

$$\{ (2,6) \quad (6,2) \quad (3,5) \quad (5,3) \quad (4,4) \}$$

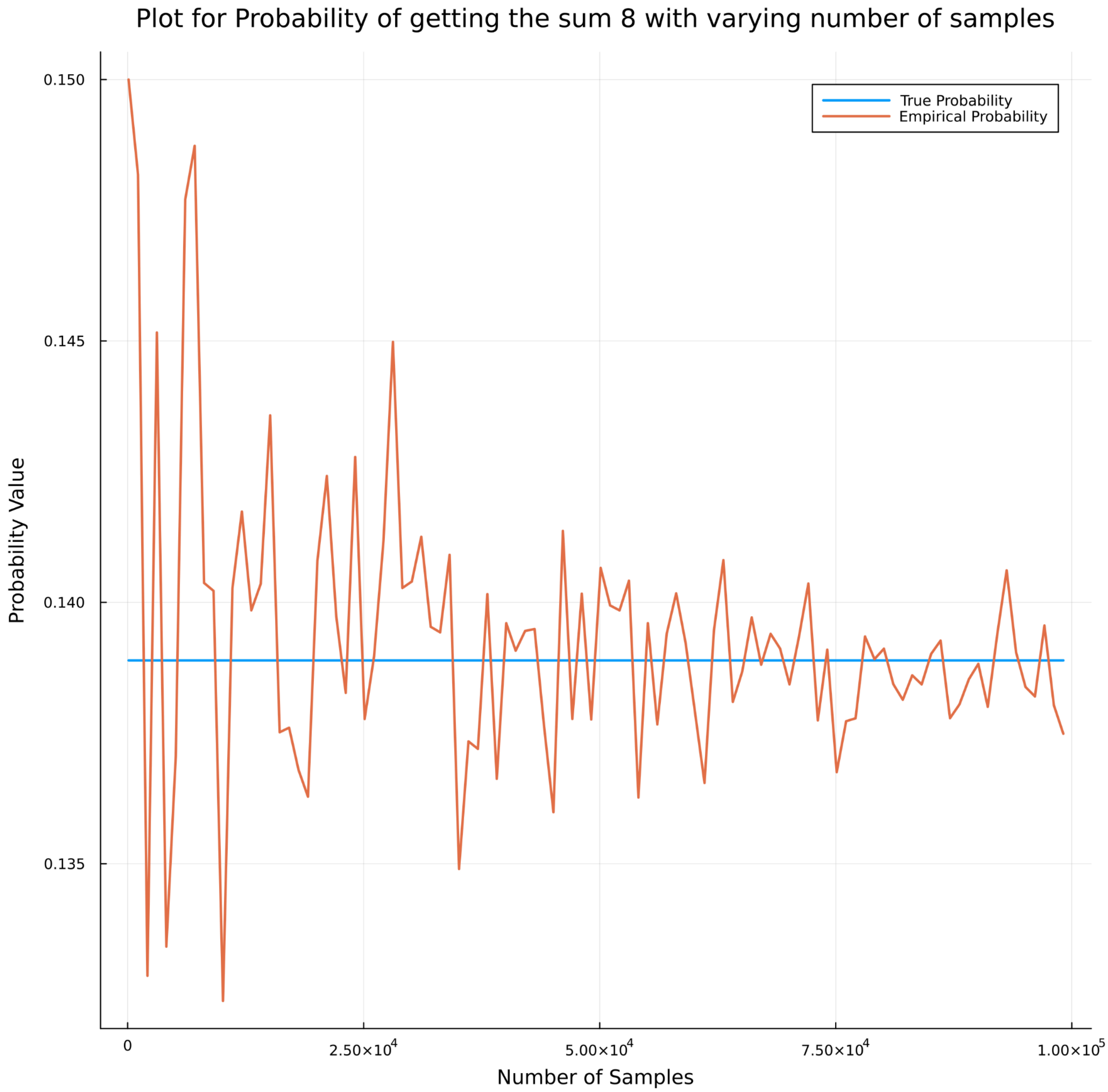
$\therefore$ Probability of getting sum of 8 $= \dfrac{5}{36} = 0.138888$

Plots for Problem 3

1) Plot when number of samples varied from 10 to 1000



2) Plot when number of samples varied from 1000 to 100000



Problem 4

1. **False.**

The KL expansion is only possible when the random process has finite variance, i.e. it is second order. However, it doesn't have to be Gaussian. It is possible for non-Gaussian stochastic processes as well.

2. **True**

From class discussion, we know that the trucated KL expansion is a finite-dimensional unbiased approximation of $X(t, \omega)$.

If $X(t, \omega)$ is WSS, then it has constant mean and so the truncated KL expansion will have a constant mean too, thus making it WSS.

3. False

No, unbiased Parameter estimators are not generally preferred. Estimators with low MSE values are preferred, and we know from class discussion that unbiased estimators can have higher MSE values than biased estimators.

4. False

The data need not be statistically independent. I.I.D data points make the calculation for the MLE estimate easier, but as long as the likelihood function can be defined for the data points, the MLE estimate can be obtained.

Problem 5

ODE model $\rightarrow$ $u_t = -ku$

Initial condition $\rightarrow$ $u(t) = u_0$

Solution $\rightarrow$ $u(t) = u_0 e^{-kt}$

$k > 0$ and $k \sim U[a \ b]$

We wish to compute mean & variance of $u(t)$ at $t = 10$

Part 1

We know

$$E[u(10)] = \int_a^b \left(u_0 e^{-10k}\right) \cdot p(k) \cdot dk$$

$$= \underline{u_0} \int_a^b e^{-10k} \cdot p(k) \cdot dk$$

and

$$Var(u(10)) = E[u^2(10)] - (E[u(10)])^2$$

$$E[u^2(10)] = u_0^2 \int_a^b e^{-20k} \cdot p(k) \cdot dk$$

We know that $e^{-\alpha x}$ is a smooth analytic function.

So, from class discussion, we know that

when f is a smooth analytic function,

$$|I(f) - \hat{I}(f)| = O(c^{-N})$$

where c is a constant

So, there is an exponential convergence with number of points N.

Also, since $\underline{k}$ is uniformly distributed, we can use the Legendre - Gauss quadrature rule to obtain N abscissas and weights.

To obtain N abscissas and weights,

we first solve the Legendre polynomial of degree N to get abscissas $y_1, y_2 \dots y_N$

and then the integrals of Lagrange interpolating polynomials to get weights $w_1, w_2 \dots w_N$

These can be easily obtained using the Golub-Welsch algorithm.

Once we have $\{(y_i, w_i)\}_{i=1}^{i=N}$, we can approximate mean and variance as follows:

$$E[u(10)] = \underline{\mu_0} \int_a^b e^{-10k} \cdot p(k) \cdot dk$$

There is a need for change of variable to ensure limits are from -1 to 1 and not a to b.

Let $k = \left(\frac{b-a}{2}\right) r + \left(\frac{b+a}{2}\right)$

$\therefore \; dk = \left(\frac{b-a}{2}\right) dr$

R.W

$k = mp + c \qquad p \to -1, 1 \quad k \to a, b$

$a = -m + c$

$b = m + c$

$\Rightarrow c = \frac{a+b}{2} \; ; \; m = \frac{b-a}{2}$

$\Rightarrow k = \left(\frac{b-a}{2}\right) p + \left(\frac{b+a}{2}\right)$

$$\therefore E[u(10)] = \mu_0 \int_{-1}^{1} e^{-10\left[\left(\frac{b-a}{2}\right) r + \left(\frac{b+a}{2}\right)\right]} \cdot p(r) \cdot \left(\frac{b-a}{2}\right) \cdot dr$$

$$= \frac{1}{2} \cdot \left(\frac{b-a}{2}\right) \cdot \mu_0 \int_{-1}^{1} e^{-10\left[\left(\frac{b-a}{2}\right) r + \left(\frac{b+a}{2}\right)\right]} \cdot dr$$

$$= \frac{u_0 (b-a)}{4} \int_{-1}^{1} e^{-10\left[\left(\frac{b-a}{2}\right) r + \left(\frac{b+a}{2}\right)\right]} \cdot dr$$

$$= \frac{u_0 (b-a)}{4} \cdot \sum_{i=1}^{N} w_i \, e^{-10\left[\left(\frac{b-a}{2}\right) y_i + \left(\frac{b+a}{2}\right)\right]}$$

Similarly,

$$E[u^2(10)] = \frac{u_0 (b-a)}{4} \cdot \sum_{i=1}^{N} w_i \, e^{-20\left[\left(\frac{b-a}{2}\right) y_i + \left(\frac{b+a}{2}\right)\right]}$$

$$\text{Var}(u(10)) = E[u^2(10)] - \left(E[u(10)]\right)^2$$

Part 2

Yes, stochastic collocation can be used to compute $E[u(10)]$ and $\text{Var}[u(10))$ when u_0 is also a Random Variable. We can use the multidimensional rules discussed in lectures since u_0 is independent of k. We can compute abscissas and weight for each dimension separately and compute the tensor product grid to obtain $E[u(10)]$ and $\text{Var}[u(10)]$

```
given_data_matrix =
[
    3.26 6.8 3.79 2.45 2.67 2.88 5.54 4.63 4.13 3.02 4.1 1.53 1.96 2.96 2.69 4.26 2.57 6.92 1.7 3.13 3.15 2.68 3.14 5.3 2.29 1.71 3.54 2.1 3.31 2.97 4.12 2.92 4.36 3.62 1.96 4.56 2.04
]

given_data = vec(given_data_matrix)
minimum_value = minimum(given_data)
start_value = floor(minimum_value)
maximum_value = maximum(given_data)
end_value = ceil(maximum_value)

num_bins = length(start_value:1.0:end_value)
counts = zeros(num_bins)

for value in given_data
    index = Int(floor(value))
    counts[index] += 1
end

bin_values = collect(start_value:1.0:end_value)
bin_probability_values = counts/length(given_data)
dist = SparseCat(bin_values,bin_probability_values)
histogram(given_data, bins=num_bins, title="Histogram of Given Data", xlabel="Bins (intervals 1 unit apart)", ylabel="Frequency", legend=false)


num_samples = 1000
generated_data = []
for i in 1:num_samples
    sampled_bin = rand(dist)
    sampled_value = sampled_bin + rand()
    push!(generated_data,sampled_value)
end

generated_data_counts = zeros(num_bins)
for value in generated_data
    index = Int(floor(value))
    generated_data_counts[index] += 1
end
generated_data_bin_probability_values = generated_data_counts/num_samples
histogram(generated_data, bins=num_bins, title="Histogram of Generated Data", xlabel="Bins (intervals 1 unit apart)", ylabel="Frequency", legend=false)

# histogram(given_data, bins=num_bins, title="Histogram of given data", xlabel="Value", ylabel="Frequency", legend=false)
```

```julia
using Random
using Plots

function sample_from_fair_die(rng)
    return rand(rng,1:6)
end


function sample_sum_of_rolls(rng)
    return sample_from_fair_die(rng) + sample_from_fair_die(rng)
end

function main()

    num_samples_array = collect(10:10:1000)
    prob_values = zeros(length(num_samples_array))

    i = 1
    for num_samples in num_samples_array
        sum_array = zeros(12)
        for i in 1:num_samples
            rng = MersenneTwister()
            sampled_sum = sample_sum_of_rolls(rng)
            sum_array[sampled_sum] += 1
        end
        prob_values[i] = sum_array[8]/num_samples
        # println(sum_array)
        i+=1
    end


    snapshot = plot(size=(900,900),
    dpi=300,
    # xticks=num_samples_array,
    # yticks=0:0.1:0.4,
    xlabel="Number of Samples",
    ylabel="Probability Value",
    title="Plot for Probability of getting the sum 8 with varying number of samples",
    # axis=([], false),
    # legend=:bottom,
    legend=true,
    )

    true_prob = 5/36
    plot!(snapshot,num_samples_array,ones(length(num_samples_array))*true_prob,label="True Probability",linewidth=2)
    plot!(snapshot,num_samples_array,prob_values,label="Empirical Probability",linewidth=2)

    display(snapshot)
    return num_samples_array,prob_values,snapshot
end

#=
num_samples_array,prob_values,snapshot = main()
savefig(snapshot,"Q3_plot.png")
=#
```