IS 605 - Assignment 9

Dan Fanelli

March 28, 2016

Problem Set 1:

This week, we'll empirically verify Central Limit Theorem. We'll write code to run a small simulation on some distributions and verify that the results match what we expect from Central Limit Theorem. Please use R markdown to capture all your experiments and code. Please submit your Rmd file with your name as the filename.

(1) First write a function that will produce a sample of random variable that is distributed as follows:

- f(x) = x; 0 <= x <= 1
- f(x) = 2-x; 1 < x <= 2

```
func_1 <- function(x){
  if(x >= 0 && x <= 1){
    return (x)
  }else if(x > 1 && x <= 2){
    return (2-x)
  }
}
```

(2) Now, write a function that will produce a sample of random variable that is distributed as follows:

- f(x) = 1 - x; 0 <= x <= 1
- f(x) = x - 1; 1 < x <= 2

```
func_2 <- function(x){
  if(x >= 0 && x <= 1){
    1-x
  }else if(x > 1 && x <= 2){
    x-1
  }
}
```

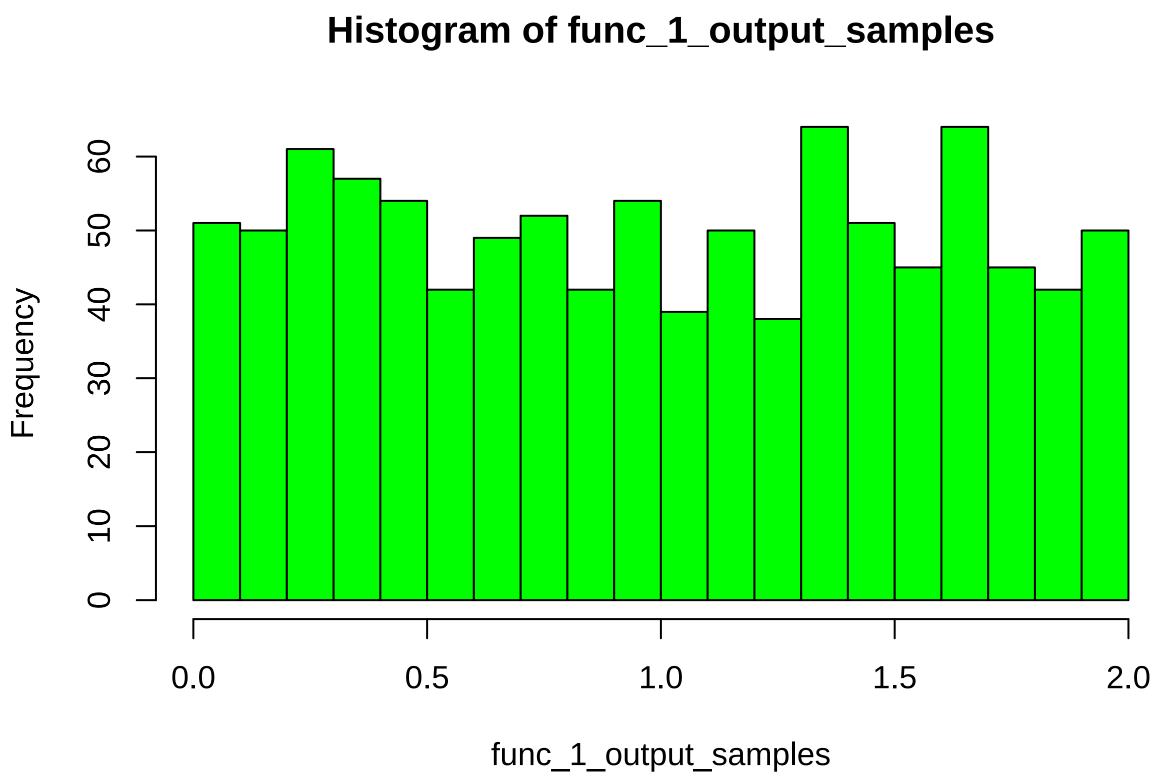
(3) Draw 1000 samples (call your function 1000 times each) from each of the above two distributions and plot the resulting histograms. You should have one histogram for each PDF. See that it matches your understanding of these PDFs. These histograms match my understanding of the 2 pdfs: the odds of any given output value are the same since both funcs are symetric from left to right, and since they are linear, the values of any given output are equal. Thus the straight-across histograms.

```r
get_sample_from <- function(the_func, size){
  inputs <- runif(size, min=0, max=2)
  outputs <- the_func(inputs)
  return (outputs)
}

SAMPLE_SIZE <- 1000
func_1_output_samples <- get_sample_from(func_1, SAMPLE_SIZE)
print(head(func_1_output_samples))
```

```
## [1] 0.46891457 0.06367588 1.35073672 0.81909001 1.73406464 0.97690222
```

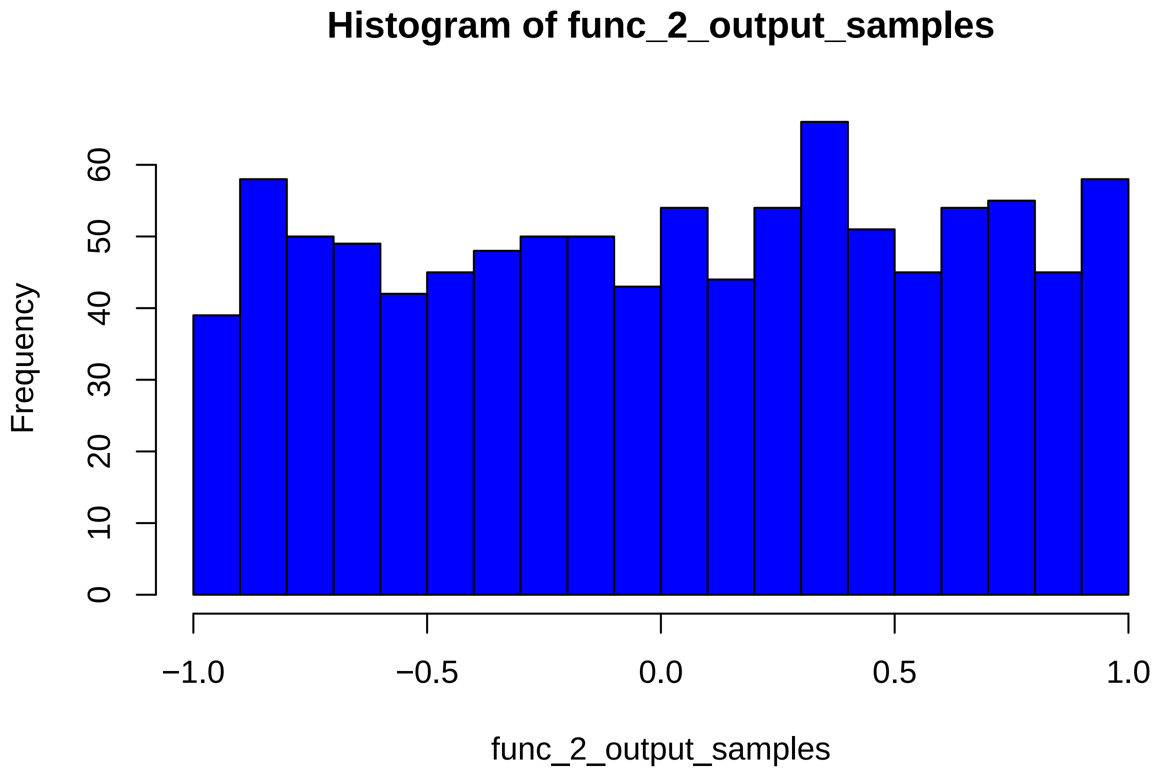
```r
hist(func_1_output_samples, col = "green", breaks = 20)
```



```r
func_2_output_samples <- get_sample_from(func_2, SAMPLE_SIZE)
print(head(func_2_output_samples))
```

```
## [1]  0.10572448 -0.75852885 -0.24794952 -0.66905562  0.09410290  0.06425625
```

```r
hist(func_2_output_samples, col = "blue", breaks = 20)
```

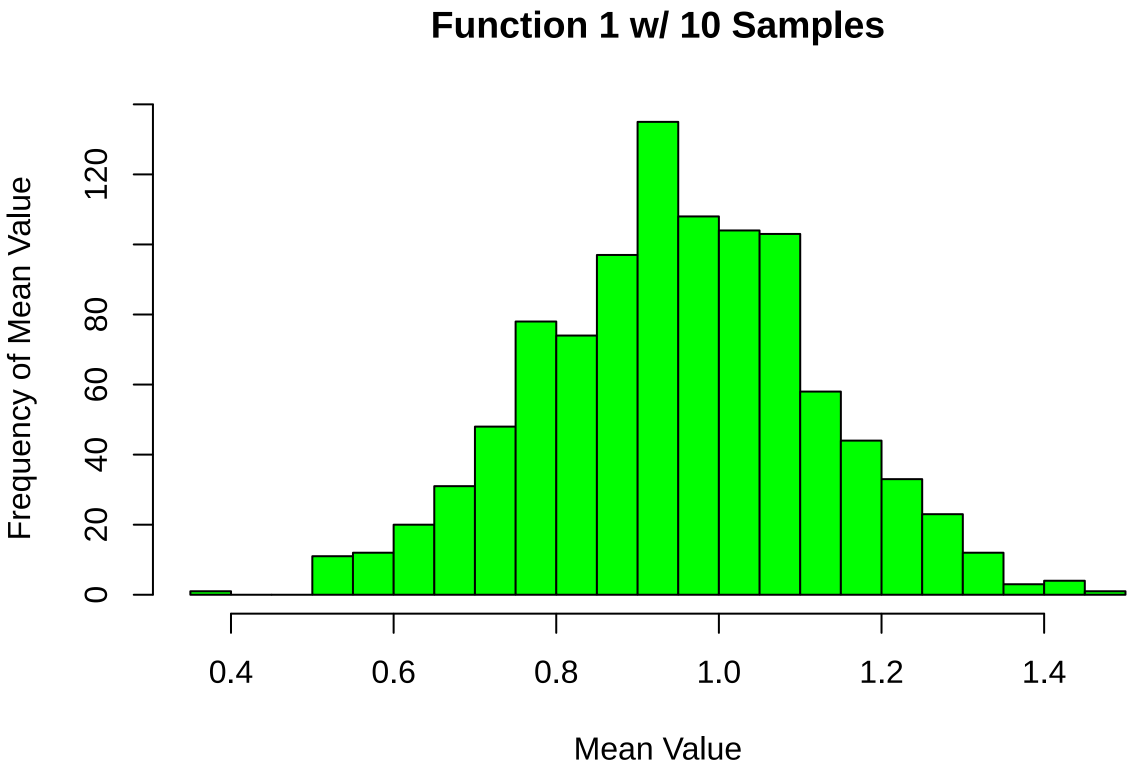


```
the_program <- function(sample_size, pdf_func, func_name, the_color){
  all_means <- numeric()
  for (i in 1:1000) {
    func_output_samples <- get_sample_from(pdf_func, sample_size)
    func_output_sum <- Reduce("+",func_output_samples)
    func_output_mean <- func_output_sum / sample_size
    all_means <- c(all_means, func_output_mean)
  }
  hist(all_means, main=paste(func_name), xlab="Mean Value", ylab="Frequency of Mean Value", col=paste(th
}
```

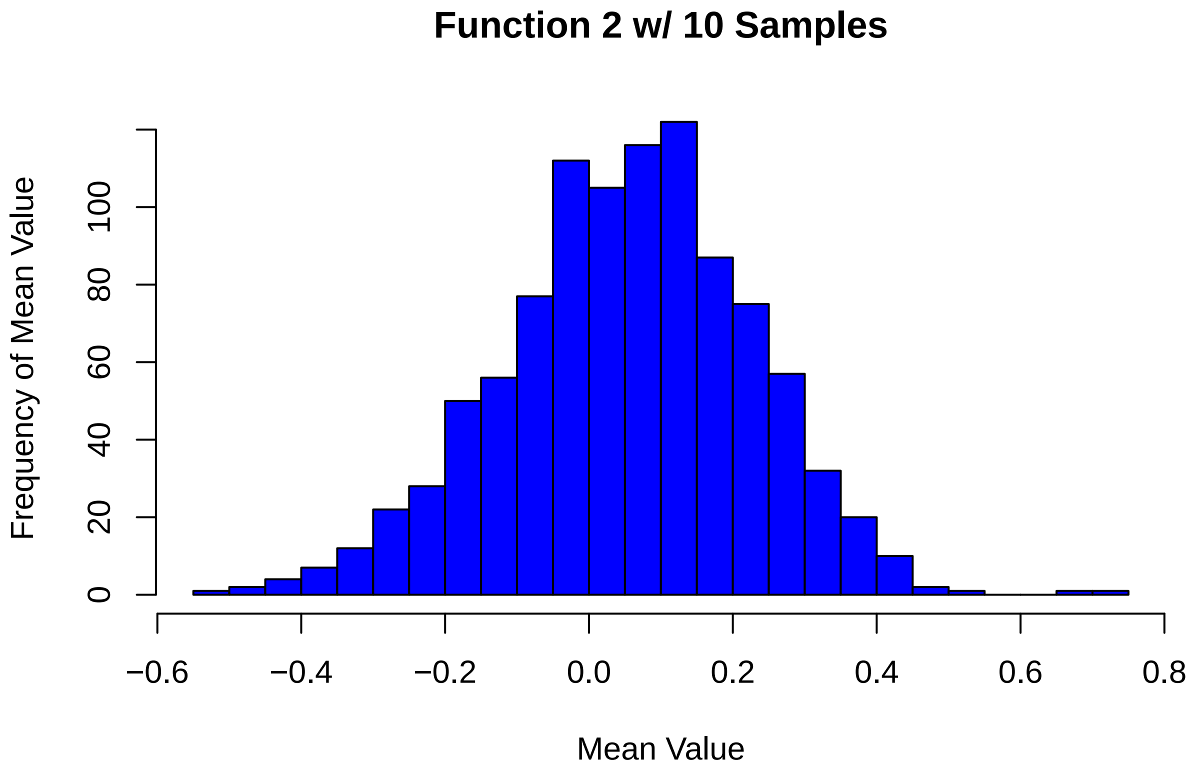
(4) Now, write a program that will take a sample set size n as a parameter and the PDF as the second parameter, and perform 1000 iterations where it samples from the PDF, each time taking n samples and computes the mean of these n samples. It then plots a histogram of these 1000 means , that it computes.

Some Results:

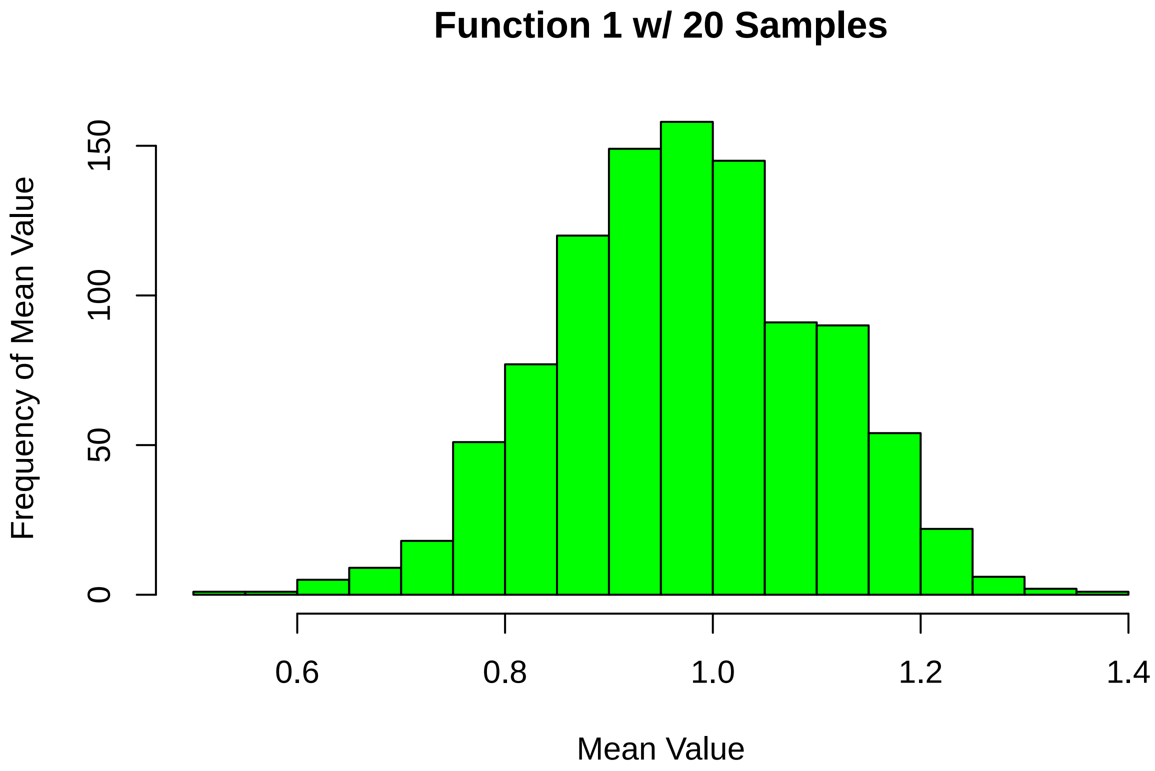
```
the_program(10, func_1, "Function 1 w/ 10 Samples", "green")
```



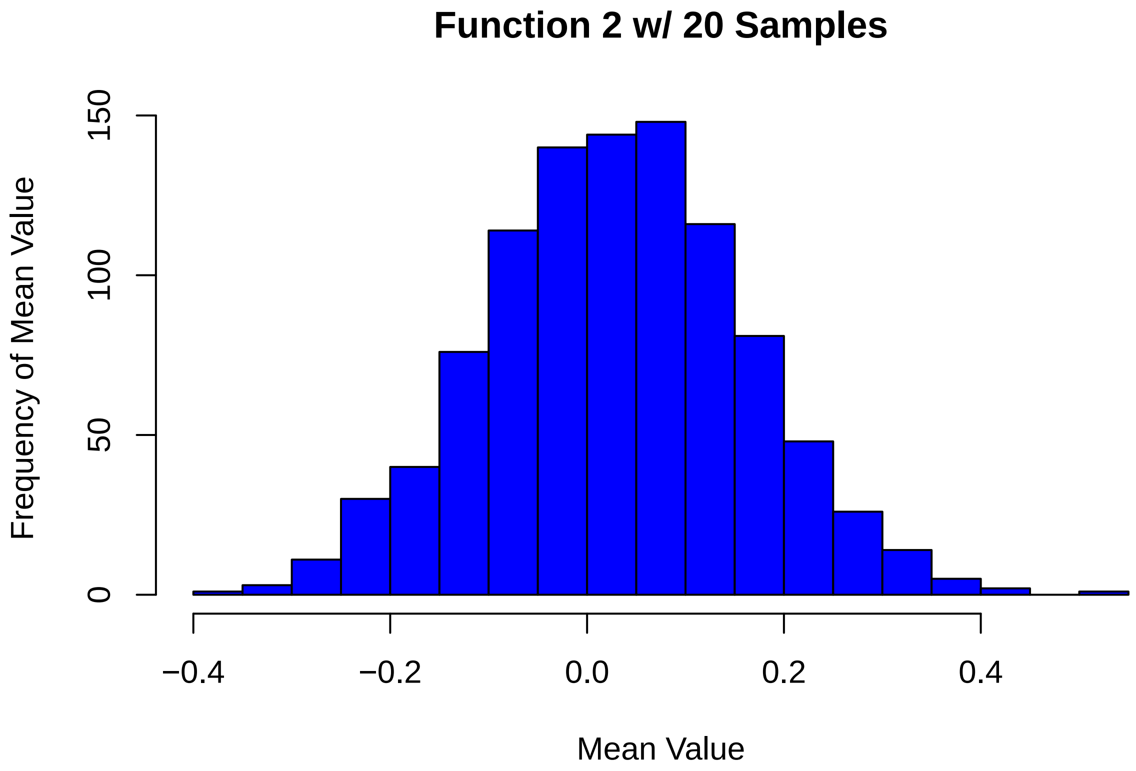
```r
the_program(10, func_2, "Function 2 w/ 10 Samples", "blue")
```



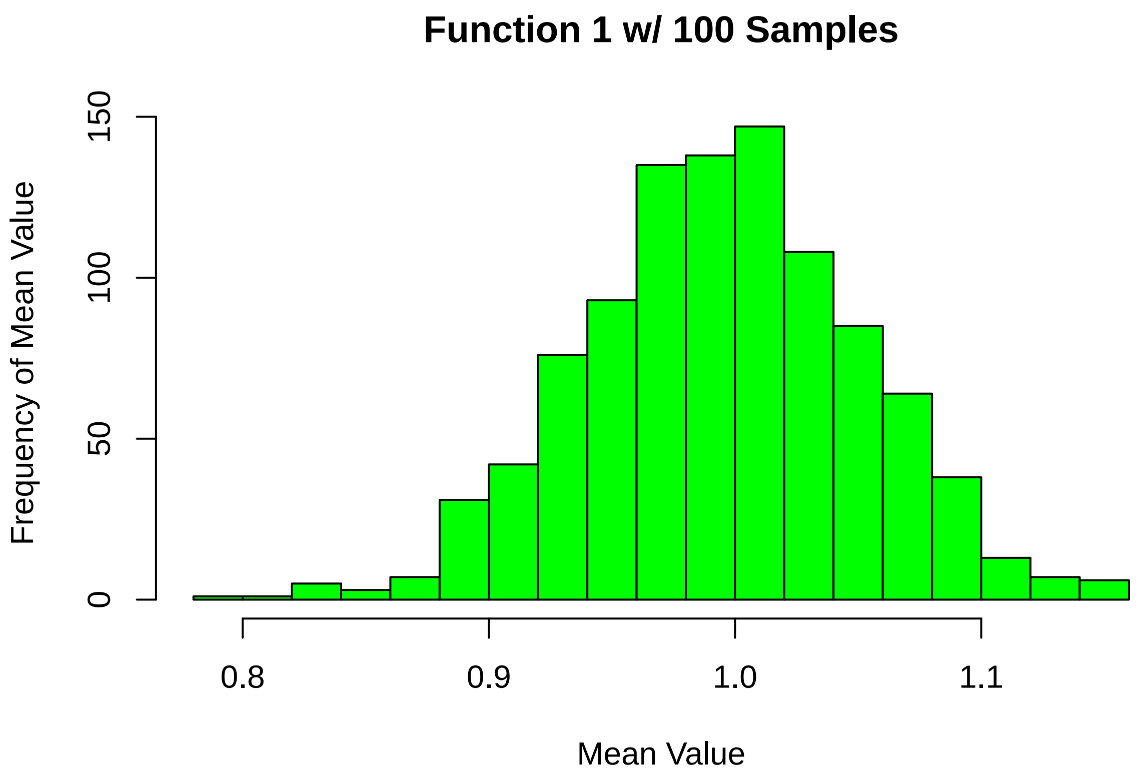
```
the_program(20, func_1, "Function 1 w/ 20 Samples", "green")
```



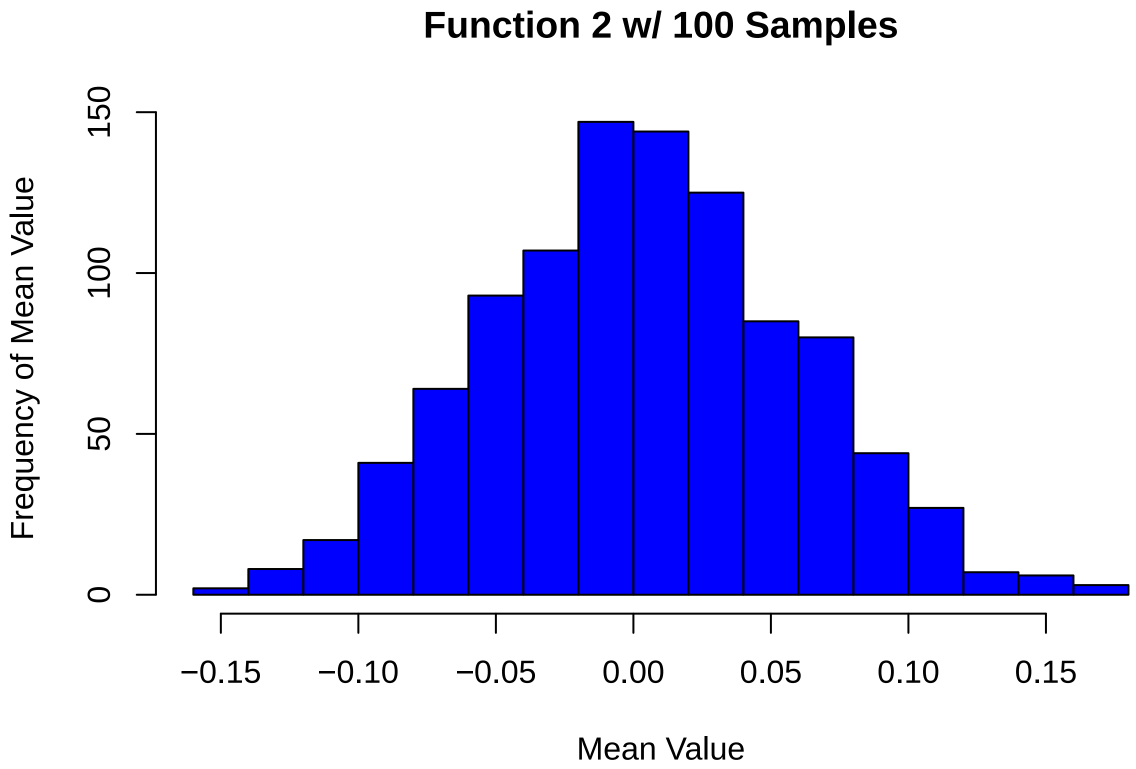
```
the_program(20, func_2, "Function 2 w/ 20 Samples", "blue")
```



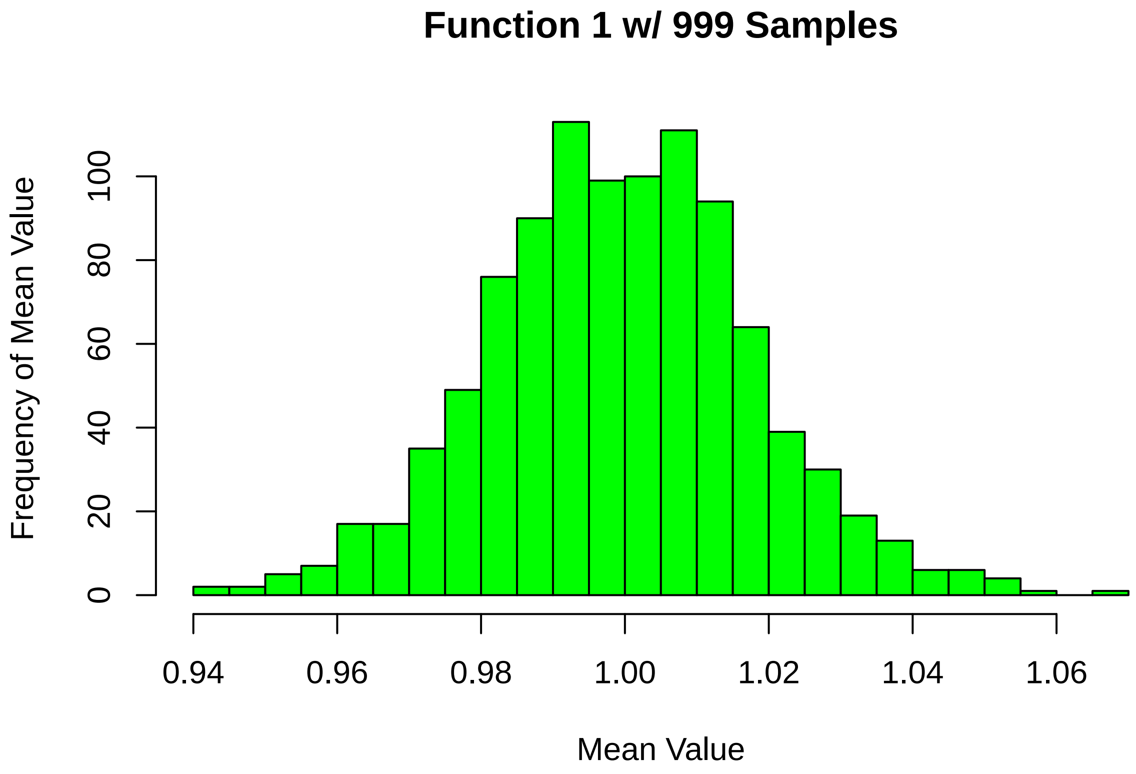
```
the_program(100, func_1, "Function 1 w/ 100 Samples", "green")
```



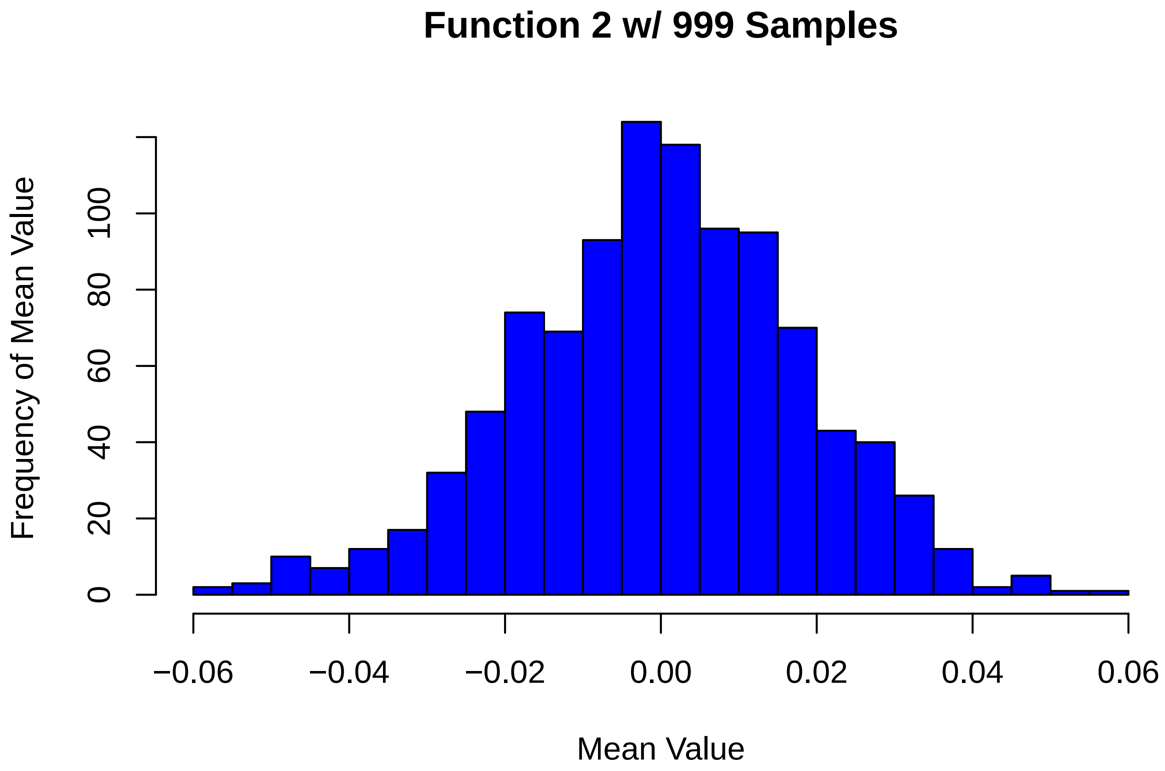
```
the_program(100, func_2, "Function 2 w/ 100 Samples", "blue")
```



```
the_program(999, func_1, "Function 1 w/ 999 Samples", "green")
```



```
the_program(999, func_2, "Function 2 w/ 999 Samples", "blue")
```


Function 2 w/ 999 Samples
Frequency of Mean Value
0
20
40
60
80
100
−0.06
−0.04
−0.02
0.00
0.02
0.04
0.06
Mean Value